

FlashCONNECT 3.5

Administrator's Guide



Copyright (c) 1982, 2004. Raining Data Corporation. All rights reserved. *Raining Data*, *TigerLogic*, *D³* (referred to as D3), *Pick*, *Omnis*, *mvDesigner*, and *mvEnterprise* are registered trademarks or trademarks of Raining Data Corporation.

This product may include or be accompanied by software developed by third parties, including those listed below. Software from third parties may be subject to additional or different license terms. For details, please see the Raining Data End User License Agreement, the third party license agreements that accompany the product, and the product's documentation.

Customers of Raining Data may find the following third party open source programs of use ("Open Source Programs"):

- Linux Operating System (only accompanies D3 ProPlus product). See source code for applicable copyright notices.
- GBD, the GNU Project Debugger. Copyright (c) 2002 Free Software Foundation.

These Open Source Programs are publicly available at no charge. As a convenience to you, Raining Data may provide you with copies of these Open Source Programs in the same package as Raining Data products. However, these Open Source Programs are not part of Raining Data's product and are not subject to the Raining Data End User License Agreement. Rather, these Open Source Programs are free software; you can redistribute them and/or modify them under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License or (at your option) any later version. These Open Source Programs are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You will find a copy of the GNU General Public License accompanying these Open Source Programs. If you do not, please contact Raining Data. For those Open Source Programs that are provided, source code should also be included on the accompanying media. If not, please contact Raining Data at any time within three (3) years after you acquire this Raining Data product to request a free copy the source code for these Open Source Programs.

FlashCONNECT Version:	3.5
Database Version:	See Installation Guide
Document Version:	November 2004

Preface

This publication is intended for a technical audience with knowledge about a database (D3 or mvBase), BASIC, some knowledge of UNIX or Windows, and moderate knowledge of HTML and the Internet.

This Preface contains:

- [Document Conventions](#)
- [Related Documentation](#)

Document Conventions

In this document, references to Windows include:

- Windows 2000
- Windows XP Professional (Pro)
- Windows 2003

In this document, references to UNIX include:

- AIX
- Linux
- SunOS (Web server components only)

All Windows dialog boxes, edit boxes, screens, and product displays in this guide are shown in Windows 2000. Some features and options may not be available or may be accessed differently on other Windows systems.

NOTE— Although the database account containing the FlashCONNECT software may be named differently for your installation, this document usually refers to it as the WWW account.

Syntax Conventions

Convention	Description
<code>monospace</code>	Indicates code examples, syntax, commands, options, and other keywords recognized by the system.
<code>monospace bold</code>	Type words or characters in monospace bold at the TCL prompt. These boldface words are commands, file names, options, and other keywords recognized by the system.
<i>Italic</i>	<ul style="list-style-type: none">• Variables to be replaced by the value. It can be an actual name, word, or number. For example, <i>file.name</i> might be replaced with CUST.FILE.• Non-linked references to other <i>documents</i> and <i>topics</i>.• Emphasis for impact or the first time a word is used. For example, ...resources and information make up the state of the application.

Convention	Description
$n\{-m\}$	The letters n and m represent numbers. Numbers are usually input as a single string without commas. Numbers can also be used to define a range designation. The starting and ending numbers are typed with a hyphen between. For example, $n\{-m\}$ could be replaced by 10 or by 1-31.
{ }	Braces ({ }) delimit a mutually exclusive set of options.
on off	The vertical line or pipe sign () separates a mutually exclusive set of options.
[0 line ?]	Brackets ([]) indicate optional entries.

File Naming Conventions

The w3... file names (w3Exec, w3HtmlInsert, and so on) use capitalization to make them easier to read in the text. The casing is not necessary when typing them in your applications.

If your database is case sensitive, all constants must be in uppercase although mixed cases may be used in our examples.

Active Links

When this guide is viewed in Portable Document Format (PDF), some text contains hypertext links to Web sites and appear as:

<http://www.rainingdata.com/products/connect/fc/index.html>

Underlined cross references are also active links that jump to that topic. These references appear like this:

See [Syntax Conventions](#) for more information.

Related Documentation

To download or order related documentation published by Raining Data Corporation, please refer to the Raining Data Web site at:

<http://www.rainingdata.com/support/documentation/index.html>

These links can also be used to access the FlashCONNECT *Programmer's Online Reference* quickly.

[Programmer's Online Reference](#)

Contents

Preface	iii
Document Conventions	iv
Syntax Conventions	iv
Active Links	v
Related Documentation	vi
Chapter 1	Introduction 1
	About This Document 2
	Topic Descriptions 2
	Document Conventions 4
	About FlashCONNECT 7
	Architecture 11
	Selecting a Web Server 12
	FlashCONNECT Application Types 13
	Static 13
	Dynamic 13
	About mvDesigner 15
	Interactive Multiuser Web Applications 16
	FlashCONNECT Sessions 16
	Session Tracking Problem 16
	FlashCONNECT Solution 17
	Explanation of Our Solution 18
	Connection Details 20
	Using Virtual Directories 22
Chapter 2	Getting Started 25
	Administrative Tasks 26
	w3Users, w3Apps, and w3Groups Files 27
	Authentication Process 27
Chapter 3	Maintaining Applications, Groups, and Users 29
	Logging On to FlashCONNECT Maintenance Pages 30

	Adding FlashCONNECT Groups	32
	Deleting FlashCONNECT Groups	33
	Adding Users to Groups	33
	Removing and Adding Users from Membership Lists	34
	Deleting FlashCONNECT Users	34
	Assigning Users and Groups to Dynamic Applications	35
	Deleting Applications.	37
Chapter 4	Administering Server Pools	39
	Using Server Pools	40
	Connecting Database Server Pools	42
Chapter 5	Accessing Administration Reports	49
	Concurrent FlashCONNECT Use Report	50
	FlashCONNECT Sessions Report	51
	FlashCONNECT Contacts Report	52
	FlashCONNECT Redirect Report	53
	FlashCONNECT Rotating Content Report	54
	FlashCONNECT Line Status Report	55
Chapter 6	Administering the FlashCONNECT Socket Server Manager . . .	57
	Accessing the FlashCONNECT Socket Server Manager	58
	Starting the FlashCONNECT Socket Server Manager	59
	Using the FlashCONNECT Socket Server Manager	62
	Adding FlashCONNECT Socket Servers	62
	Stopping FlashCONNECT Socket Servers.	63
	Terminating FlashCONNECT Socket Server Connections	64
	Using the FlashCONNECT Socket Server Log	65
	Clearing Monitor Information	70
	Exiting FlashCONNECT Socket Server Manager	70
Chapter 7	Understanding FlashCONNECT Java Components	71
	FCServlet	72
	JNI Interface	72
	Working with FCServlet.	73
	Testing FCServlet.	74
	Test 1	74
	Test 2	75
	Using FCServlet	77
	FCServlet Diagnostic	78
	Using the FCServlet Diagnostic	78
Chapter 8	Configuring, Starting and Stopping FlashCONNECT Components .	85
	Turnkey Installation	86
	Starting FlashCONNECT in the Foreground	88
	Running FlashCONNECT Ports in the Foreground (D3 Only)	88

	Customizing the Local Environment	89
	Environment tab	89
	Startup Tab	90
	FlashCONNECT Service	92
Chapter 9	Customizing the FlashCONNECT Server and Web Server	95
	Using WWW-STATUS	96
	Explanation of WWW-STATUS and w3Lines Status Messages	98
	FlashCONNECT Heartbeats	101
	Customizing WWW-IRUN	103
	Windows Server Administration	104
	Web Server	104
	Customizing Internet Information Server (IIS), Netscape and Apache	
	Web Server Error Messages	104
	Redirecting Error Messages	105
Chapter 10	Administering Log and JOBS Files	109
	Administering Log Files	110
	Administering the JOBS File	112
Chapter 11	Using Character Translation	115
	Performing Character Translation	116
	1 to 1 Translation	116
	1 to 2 and 2 to 1 Translation	116
	Enabling Character Translation	117
	Deleting Translation Server Pools	119
	Character Translation Error Messages	120
Appendix A	FlashCONNECT SOAP Access	121
	Deploying the FlashCONNECT SOAP Interface	122
	Deploying the execute Interface Methods	123
	Generating a wsdl File for Client Software (Optional)	124
	Creating the FlashCONNECT Database Application	127
	Accessing the execute Interfaces from Client Programs	128
Appendix B	Debugging and Diagnosing FlashCONNECT Configuration	137
	Diagnosing FlashCONNECT Configuration Problems (D3 Only)	138
	Running the Loop-Back Test (D3 Only)	143
	Roving Tandem Debugger	144

	Application Debug Strategies	146
	Static Applications	146
	Dynamic Applications	146
	Determining if the Application is in the Debugger	147
	Using the FlashCONNECT Server Debug Program	147
	Debugging Server Pools	150
	Debugging w3Monitor	151
	Recording Outgoing HTML Code and Incoming Variables	151
	Testing FCSservice on a Specified Machine and Port	153
	Logging Diagnostic Messages for FCSservice and FCsocketServer	153
	Examining CGI and Form Variables	155
	Diagnostic Messages Displayed On the Browser	155
	Determining Release Levels	157
Appendix C	Troubleshooting for UNIX Platforms	159
	Troubleshooting for the UNIX Platform.	160
	Diagnostics	160
	Using the fccgi.pl trace Utility	167
Appendix D	Troubleshooting for Windows Platforms	169
	Troubleshooting for Windows	170
	Diagnostics	170
	Using the fccgi.pl trace Utility	177
	FlashCONNECT Event Log Entries	178
Appendix E	MSG_Exception Messages	181
	MSG_Exception: Event Handler Messages	182
Appendix F	Error Messages	205
	UNIX Platform Error Messages	206
	Windows Platform Error Messages	208
Appendix G	Customer Service	211
	Contacting Customer Service	212
	Providing Required System Information	212
	Receiving Extended Support Services	212
	Purchasing Support Packages	212
	Activating Your Software	212
Index	213

Introduction

This publication is intended for administrators knowledgeable about:

- D3, mvBase, and Pick BASIC
- UNIX or Windows operating systems
- HTML and the Internet

All topics apply to both the UNIX and Windows platforms, unless a specific platform is specified.

We hope this guide is a valuable tool for administering and maintaining the FlashCONNECT software.

About This Document

The purpose of this publication is to explain FlashCONNECT system administration.

NOTE— The FCService components are necessary only for FlashCONNECT dynamic applications and mvDesigner.

Topic Descriptions

Use the table below as a guide to this document.

Type	Topic	Description
Chapter 1	Introduction	Discusses key concepts and explains how FlashCONNECT works.
Chapter 2	Getting Started	Discusses the steps to take after installing FlashCONNECT.
Chapter 3	Maintaining Applications, Groups, and Users	Provides instructions for setting up, adding and deleting FlashCONNECT users and groups.
Chapter 4	Administering Server Pools	Explains how to connect, start, and stop server pools, as well as load balancing and manual control.
Chapter 5	Accessing Administration Reports	Provides instructions for viewing HTML-based reports along with examples.
Chapter 6	Administering the FlashCONNECT Socket Server Manager	Discusses how to set up, start, and stop FlashCONNECT Socket servers.
Chapter 7	Understanding FlashCONNECT Java Components	Describes the Java Native Interface, Java Servlets, and Java Virtual Machine Environment.

Type	Topic	Description
Chapter 8	Configuring, Starting and Stopping FlashCONNECT Components	Provides turnkey installation instructions, customizing the local environment, starting and stopping FCService, and starting FlashCONNECT with D3.
Chapter 9	Customizing the FlashCONNECT Server and Web Server	Discusses FlashCONNECT heartbeats, customizing WWW-IRUN and unnamed Common.
Chapter 10	Administering Log and JOBS Files	Provides instructions for administering log and JOBS files and determining release levels.
Chapter 11	Using Character Translation	Describes how to use character translation.
Appendix A	FlashCONNECT SOAP Access	Discusses deploying FlashCONNECT for SOAP (Simple Object Access Protocol) access.
Appendix B	Debugging and Diagnosing FlashCONNECT Configuration	Provides applications for diagnosing configuration problems, running the Loopback test, application debug strategies, troubleshooting for UNIX platforms, troubleshooting for Windows platforms, and understanding error messages.
Appendix C	Troubleshooting for UNIX Platforms	Contains strategies for troubleshooting problems associated with UNIX platforms.
Appendix D	Troubleshooting for Windows Platforms	Contains strategies for troubleshooting problems associated with Windows platforms.
Appendix E	MSG_Exception Messages	Provides and describes MSG_Exception error messages.
Appendix F	Error Messages	Provides and defines common error messages that you might encounter.
Appendix G	Customer Service	Provides Customer Service contact information.

Document Conventions

Different typefaces and type styles are used throughout this guide to indicate specific kinds of information.

-
- NOTE—**
- Although the database account containing the FlashCONNECT software may be named differently for your installation, this document usually refers to it as the `www` account.
 - In some cases where a software change will be implemented soon, you may notice that this document differs from actual displays in your specific installation.
-

Typefaces

The regular text of this document appears in a typeface like this:

This topic explains how to...

Words used to indicate specific D3 words and to indicate a dialog between the computer and the user, appear in a monospace type like this:

Enter the TCP/IP port `FCService` is using

When this guide is viewed in PDF format (Portable Document Format), certain text contains hypertext links to the Raining Data Web site. These links can be used to access the FlashCONNECT *Programmer's Online Reference* quickly. These hypertext links will appear like this:

<http://www.rainingdata.com/support/documentation/fc/Index.html>

[Programmer's Online Reference](#)

Cross references indicate links to other topics in any of the FlashCONNECT documents. These links appear like this:

See [Document Conventions](#) for information on typeface usage.

Entering Data

Action	Description
Enter	Input the specified commands or text as shown in the instruction and then press the carriage return key, usually labeled as <Enter>, <Return>, or <New Line> on your keyboard.
Type	Input the specified commands or text as shown in the instruction. Do not press the carriage return key unless instructed.
Press	Press the single specified character or combination of characters as instructed.

Syntax Conventions

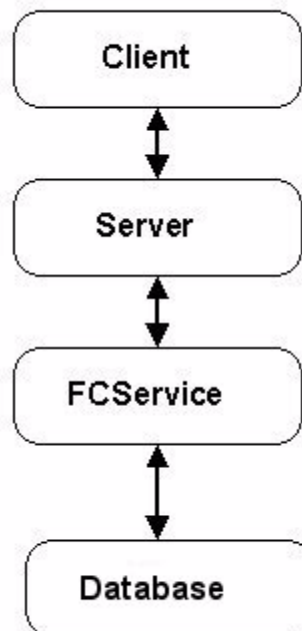
Convention	Description
<code>monospace</code>	Indicates code examples, syntax, commands, options and other keywords recognized by the system.
<code>monospace bold</code>	Enter (type) words or characters in <code>monospace bold</code> at the TCL prompt. These boldface words are commands, options, and other keywords recognized by the system.
<i>Italic</i>	<i>Italic</i> text indicates: <ul style="list-style-type: none"> Variables to be replaced by the value. It might be an actual name, word, or number. For example, <i>baud_rate</i> might be replaced by 9600. Emphasis for impact or the first time a word is used. For example, "These resources and information make up the <i>state</i> of the application."
<i>n{-m}</i>	The letters <i>n</i> and <i>m</i> represent numbers. Numbers are usually input as a single string without commas. Numbers may also be used to define a range designation. The starting and ending numbers are typed with a hyphen between. For example, <i>n{-m}</i> might be replaced by 10 or by 1-31.
{ }	Braces ({ }) delimit a mutually exclusive set of options.
on off	The vertical line or pipe sign () separates mutually exclusive options.
[0 line ?]	Brackets ([]) indicate optional entries.

File Naming Conventions

The w3... file names (w3Exec, w3HtmlInsert, and so on) use capitalization to make them easier to read in the text. The casing is not necessary when typing them in your applications.

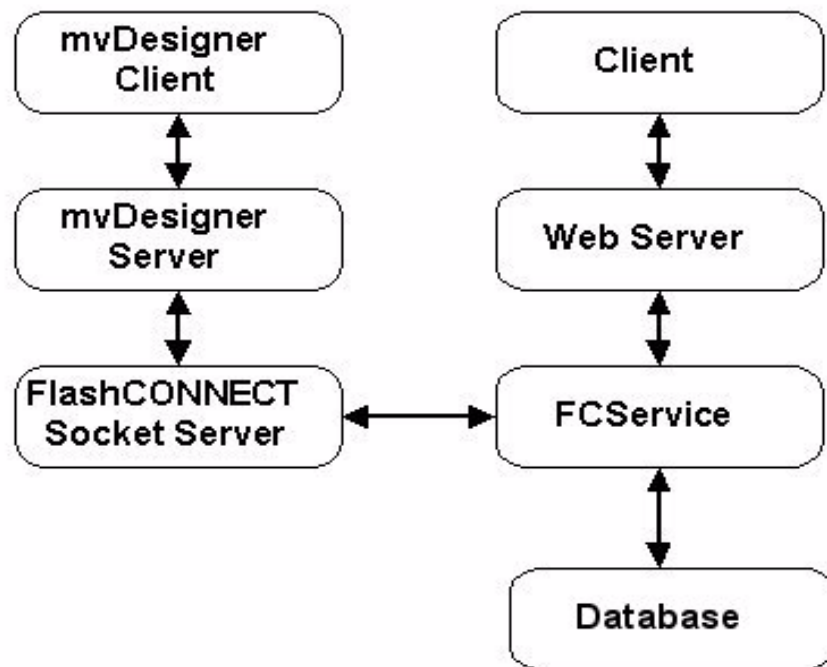
About FlashCONNECT

Earlier releases of FlashCONNECT provided a browser-based middleware component between FlashCONNECT clients, Web servers and the D3 or mvBase database. This enabled remote, cross-platform capabilities for programmers to write and post applications using Pick BASIC subroutines and to display these applications in HTML templates, creating viewable Web pages.



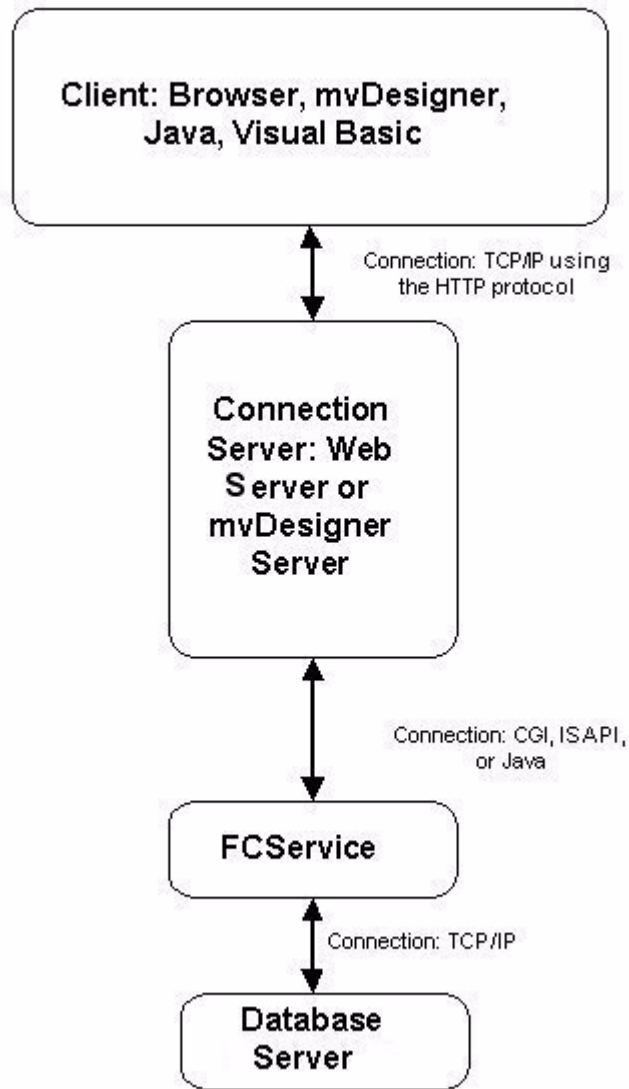
FlashCONNECT allows Pick BASIC applications (Pick BASIC business rules), via built-in subroutines, to use the external components as an input and output device. Additionally, Java, C++, Visual Basic, and other popular programming applications can work with these Pick BASIC business rules.

FlashCONNECT extends this functionality and creates more flexibility by introducing new components such as interoperability with the mvDesigner client and server through the FlashCONNECT Socket Server connections and by introducing FCServlet which increases Web server performance.



FlashCONNECT is a middleware package that connects Pick BASIC business rules with TCP/IP-based external components such as Web servers, Java programs, Javascript, Visual Basic programs and mvDesigner servers.

FlashCONNECT uses either CGI, ISAPI, Java Servlet or a TCP/IP connection to an mvDesigner server. FlashCONNECT also uses named pipes for inter-process communication. FlashCONNECT subroutines send a Web page to the user's browser. When a user submits a response, FlashCONNECT subroutines retrieve the information submitted by the user and the process repeats.



FlashCONNECT's connection to a browser or to an mvDesigner server provides a graphical user interface for your database application and meets many of today's e-commerce requirements by offering these features:

- Transparent interoperability with mvDesigner, a connection between the database and an mvDesigner server that provides a graphical user interface.
- Compatible with browsers and Web servers.
- Compatible with third-party tools such as Microsoft Front Page, Macromedia Dreamweaver and Javascript.

- Uses familiar Pick BASIC programming subroutines to get you up and running quickly.
- Creates static and active (or dynamic) Web pages from database queries.
- Supports Secure Socket Server Layers.
- Enhances Web server performance with FlashCONNECT Java Components.

Before installing FlashCONNECT, think about the features you want to use and how you want to use this product. The Java Runtime Environment must be installed on your computer in order to use FlashCONNECT's Java components with:

- FCServlet to use the Java servlet interface.
- The mvDesigner interface to provide a graphical user interface between D3, mvBase, and the mvDesigner server.

NOTE— If you do not want to use FCServlet or mvDesigner, the Java Runtime Environment is not needed.

FlashCONNECT installs on a Web server or mvDesigner server, and connects the Web or mvDesigner servers to one or more database servers. Your Web applications, whether static or dynamic, allow you to determine the type of user interaction you require.

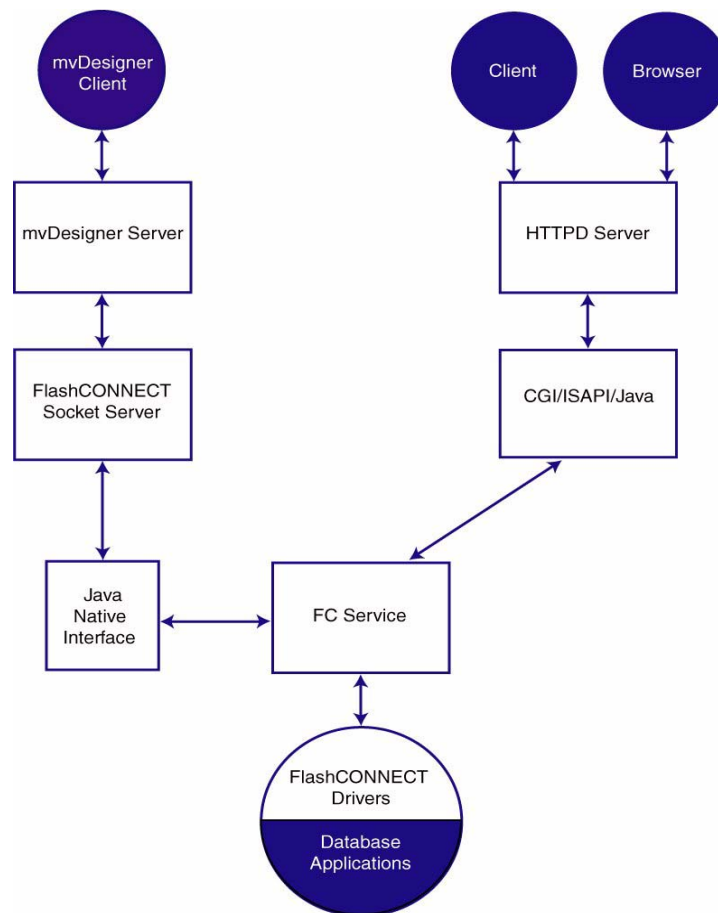
These applications can be distributed across several systems through optional server pool connections—that also allows for load balancing and tighter control of resource allocation. Additionally, creating virtual directories allows these applications and servers to communicate across multiple platforms regardless of which directory is specified in the URL or link, or which type of Web server the Web application is running on.

The following topics provide detailed information about FlashCONNECT:

- [Architecture](#) provides an illustration of the FlashCONNECT components.
- [Selecting a Web Server](#) provides Web server configuration information.
- [FlashCONNECT Application Types](#) to determine which application suits your requirements.
- [Interactive Multiuser Web Applications](#) to learn more about FlashCONNECT's session tracking model.
- [Using Virtual Directories](#) to learn how FlashCONNECT uses virtual directories.
- [FlashCONNECT Sessions](#) describes session tracking.

Architecture

The diagram below illustrates the fundamental components of FlashCONNECT and mvDesigner.



Selecting a Web Server

Almost any Web server can be used as long as it supports the CGI and ISAPI interface, which most Web servers do. The FlashCONNECT product has been tested with:

- UNIX with Netscape and Apache Web servers
- Windows 2000 IIS 5.0/XP IIS 5.1/2003 IIS 6.0

FlashCONNECT Application Types

There are two types of applications that can be built with FlashCONNECT:

- *static* (also referred to as inactive)
- *dynamic* (also referred to as active)

The table at the end of this topic provides comparisons.

Static

A FlashCONNECT application that builds static Web pages does so without processing input from the user. When static Web pages are built from a rarely changing set of data, static FlashCONNECT applications are the best choice.

For example, a static application might generate a catalog of products being sold. Since there is no need for user input, these pages are the least taxing on the underlying operating system and database, and the Web site can be updated from the database during low activity periods.

Dynamic

A dynamic application generates Web pages containing information from a database from a user interaction. A dynamic application may require persistent resources. Persistence is the capability for resources to remain allocated, and information to remain available to a specific application while Web pages are processed. These resources and information make up the *state* of the application and fall into two categories:

- Persistent data—memory, common and BASIC application, and variables
- Persistent connections—peripheral devices, record locks, and item locks

This type of use shows the strengths of FlashCONNECT dynamic applications. FlashCONNECT, not the BASIC programmer, keeps track of the state of Web pages. From the programmer's point of view, dynamic applications look very much like

BASIC applications. This allows the BASIC programmer to focus on writing an application, not on managing Web pages.

Question	Static	Dynamic
What limits the simultaneous user count?	httpd server	httpd, database server, total database licenses, number of phantoms, or available number of FlashCONNECT processes
What is the relative speed?	Faster	Fast
Is user authentication available?	No	Yes
Is information from the user or the Web environment available?	No	Yes
Are the w3Html functions available?	Yes	Yes
Can the user's input be immediately processed by an application?	No	Yes
Can cookies be set?	Yes	Yes
Can cookies be evaluated?	No	Yes

About mvDesigner

mvDesigner (a separate product from FlashCONNECT) is an application development environment used with the mvDesigner client and server.

FlashCONNECT includes the mvDesigner API that allows Pick BASIC applications to work with mvDesigner; either by driving mvDesigner screens from D3 or by encapsulating FlashCONNECT business rules in mvDesigner applications.

mvDesigner is designed for FlashCONNECT and Pick BASIC programmers to quickly create or convert Pick BASIC applications using mvDesigner's visual objects. Using mvDesigner, programmers can:

- Retrieve information sent by mvDesigner applications.
- Send data to the mvDesigner application.
- Change the form displayed by the mvDesigner application.
- Produce pop-up message boxes.
- Write persistent and non-persistent applications.

The mvDesigner API documentation is located in the FlashCONNECT [Programmer's Online Reference](#).

Interactive Multiuser Web Applications

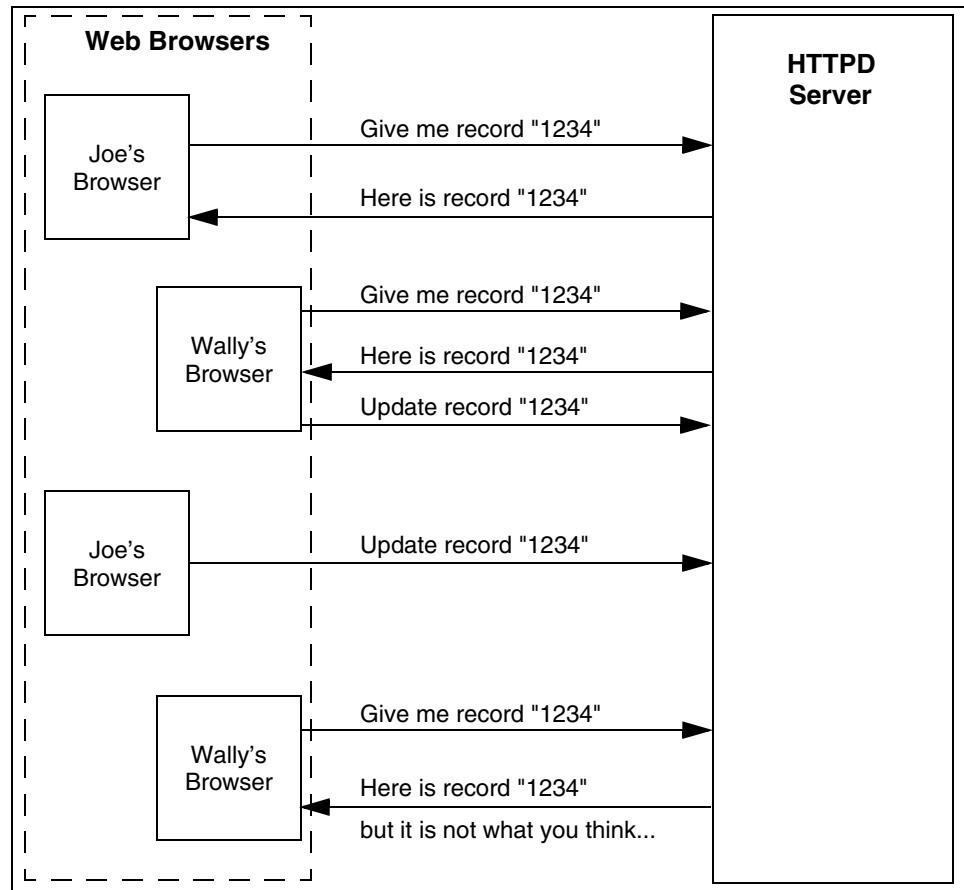
FlashCONNECT Sessions

A session begins when a user reaches the first form. At that time, the FlashCONNECT connection is available only to the browser that started the session. Unless initialized by the application, the FlashCONNECT logon screen does not start a session.

A session ends when the FlashCONNECT application returns to the FlashCONNECT application that called it. When the session ends, the FlashCONNECT connection is available for use by any authorized user.

Session Tracking Problem

The language of the World Wide Web is Hypertext Markup Language (HTML). The Hypertext Transfer Protocol (HTTP) is where the World Wide Web is based. HTTP is a stateless protocol, meaning it does not track information about successive communications. See the session tracking problem illustrated below.

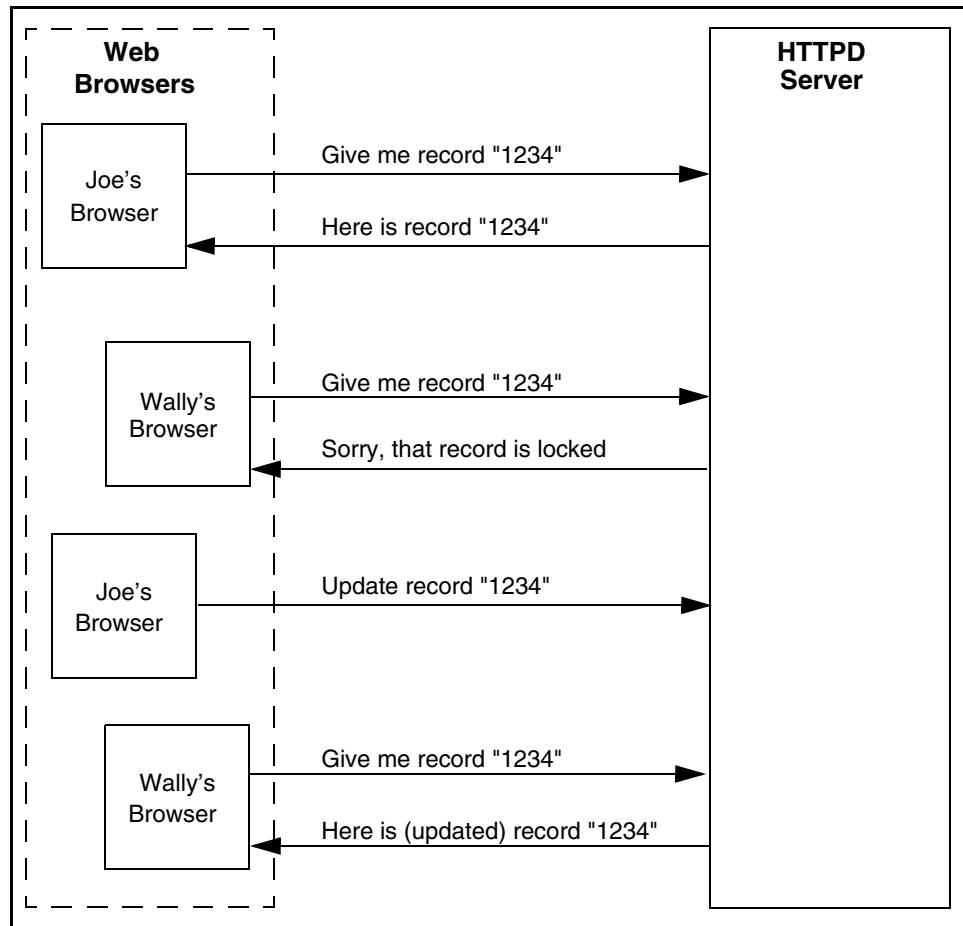


Clicking a browser link sends a message to the server. The server responds by sending the document. Once the browser receives the entire document, the server no longer tracks the session.

This works well for requesting a static page from the server; however, HTTP is not sufficient to program a multiuser interactive application. For example, consider an interactive customer update application. The server must be able to store the current state of the progressing transaction between data transfers. The server, like any application, must not allow other users to update records that are currently locked. HTTP cannot accomplish this task by itself.

FlashCONNECT Solution

FlashCONNECT resolves this session tracking problem by keeping track of the application state. See the FlashCONNECT session tracking solution illustrated below.

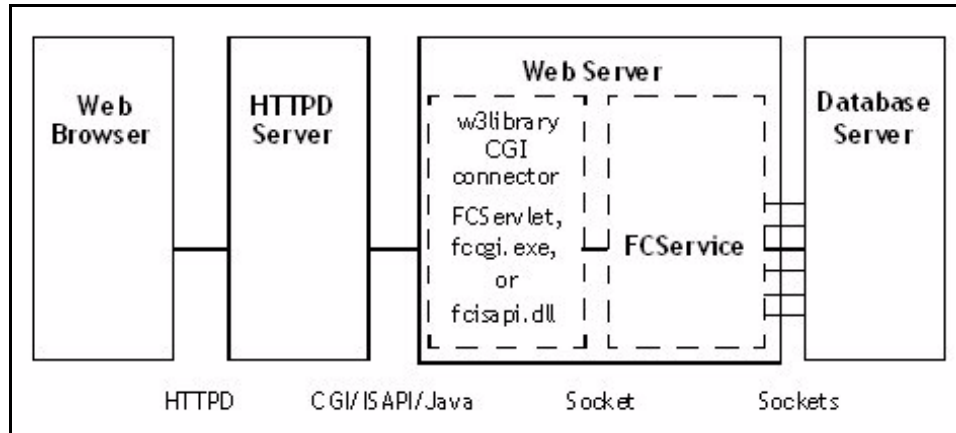


Explanation of Our Solution

In addition to the Web browser and the HTTPD server, the FlashCONNECT solution includes two elements:

- Web server (includes the FlashCONNECT CGI or ISAPI connector and FCService)
- Database server

A detailed explanation of each element, along with the FlashCONNECT solution, is illustrated in the example below.



Web Server

The Web server includes the FlashCONNECT CGI, ISAPI, and Java Servlet connectors and FCService.

fccgi.exe, fcisapi.dll, or FCServlet	<p>UNIX: The HTTPD server starts the connector application fccgi.exe or FCServlet when it receives an HTTP connection destined for a FlashCONNECT dynamic application. The connector passes the data between FCService and the HTTPD server.</p> <p>Windows: The HTTPD server starts the connector application fccgi.exe, fcisapi.dll, or FCServlet when it receives an HTTP connection destined for a FlashCONNECT dynamic application. The connector passes the data between FCService and the HTTPD server.</p>
FCService	<p>The FCService directs data packets between the Web Server or mvDesigner Server and the database server. When a connection is made, the FCService sees if the connection is part of an already established session.</p> <ul style="list-style-type: none"> • If it is a new connection, FCService begins a new session, selects a free database server socket, and passes the data from the browser to the database server. • If it is an existing connection, FCService directs the data from the browser to the database server socket that is waiting for it.

Database Server

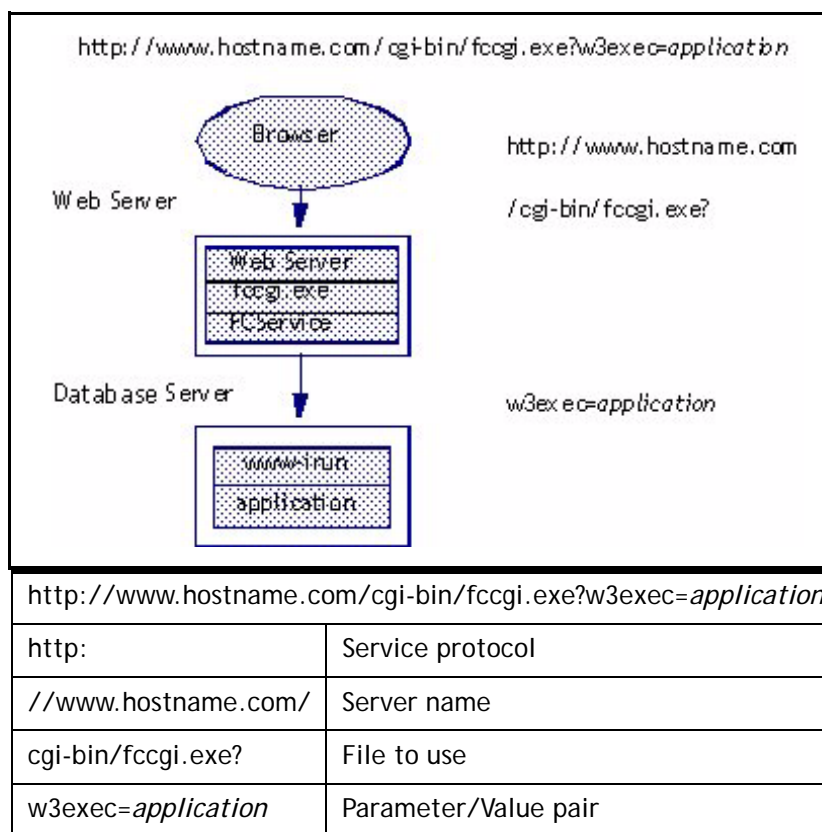
FlashCONNECT consists of an API and subroutines that implement the API. On the database server, an application runs as a called subroutine under the FlashCONNECT driver, `WWW-IRUN`.

Before the connection is made, authorization to use the application is authenticated where applicable. Once authenticated, a new session begins. See the FlashCONNECT Administrator's Guide for more information.

A complete application session can consist of several Web pages, including several connections, allowing a BASIC programmer to write software the same way as writing for a terminal.

Connection Details

This diagram describes each component and its role in the process of satisfying a request for a FlashCONNECT Web page.



This example continues using the URL in the diagram above. When a URL is entered into the browser the following occurs:

- Address portion of the URL (`http://www.hostname.com/`) is used to contact the Web server at that address and requests the specified page (`/cgi-bin/fccgi.exe?`).
- Web server processes the request and, since the requested page is an application, executes the application.
- Web server starts `fccgi.exe`, `fcisapi.dll`, or `FCServlet` which read and parse the remaining information from the URL (`w3exec=application`) and any submitted Web form. `fccgi.exe`, `fcisapi.dll`, or `FCServlet` then connect to `FCService`, passing in the environment variables, including the value of `w3exec`.
- `FCService` sees that a value for `w3exec` is specified, so it selects an unused connection to the database server and passes the information from the URL and Web form to `WWW-IRUN`.
- `FCService` sends the information to `WWW-IRUN` and starts the application specified by the value of `w3exec`.

The process to return information to the browser is:

- Application uses various FlashCONNECT API calls to create a Web page. When the application returns to `WWW-IRUN` or calling `w3Input`, the Web page created by the application is returned to `FCService` and FlashCONNECT notifies `FCService` whether the connection is persistent or not.
- `FCService` passes the information back to `fccgi.exe`, `fcisapi.dll`, or `FCServlet`. If the connection is not persistent, the connection is made available for another user. If the connection is persistent, the connection is reserved and the timeout period is started.
- `fcisapi.dll`, `fccgi.exe`, or `FCServlet` passes the page back to the Web server and finishes.
- Web server passes the page back to the browser and finishes.
- Browser displays the new Web page.

Using Virtual Directories

A virtual directory is an alias for another directory, which is an additional name for a directory that actually exists on the disk.

Virtual directories are used in FlashCONNECT in two ways:

To maintain a consistent environment between the Windows Web servers and the UNIX Web servers.	Windows Web server , IIS, creates a folder called scripts. This folder is located in the Inetpub directory (\Inetpub\scripts). (Applications or scripts are run from this directory on IIS.) A virtual folder called cgi-bin can be created to point to the scripts folder. UNIX Web servers (such as Apache) applications or scripts are run from the cgi-bin directory. A virtual directory called scripts can be created to point to the cgi-bin directory.
To use the online examples, the Web server needs access to the information provided in the WebContent folder provided with FlashCONNECT.	Create a virtual directory called w3library to point to the WebContent directory to allow the browser access to this information. On UNIX platforms, these steps are done for you as part of the standard installation process.

Virtual directories are used in FlashCONNECT in two ways:

- To maintain a consistent environment between the Windows Web servers and the UNIX Web servers. This way, regardless of which directory is specified in the URL or link, or which type of Web server the Web application is running on, the pages will work.

The Windows Web server, IIS, creates a directory called scripts. This directory is usually located under the Inetpub directory (\Inetpub\scripts). On IIS, applications or scripts are run from this scripts directory. On UNIX Web servers, such as Apache, applications or scripts are run from the cgi-bin directory.

- **Windows** Web servers, a virtual directory called cgi-bin can be created to point to the scripts directory.
- **UNIX** Web servers, a virtual directory called scripts can be created to point to the cgi-bin directory.
- To use the online examples, the Web server needs access to the information provided in the WebContent directory provided with FlashCONNECT. There are two ways to allow the browser access to this information.
 - Create a virtual directory called w3library to point to the WebContent directory.

- Copy the WebContent directory and place it under the document root directory. Rename the WebContent directory w3library.

On UNIX platforms, these steps are done for you as part of the standard installation process.

Getting Started

Once FlashCONNECT is installed, if you intend to use secured applications, you must:

- set up groups and users.
- change any configuration settings if you do not want to use the default settings.
- configure server pools.
- configure FlashCONNECT socket servers.

Administrative Tasks

The following is a list of FlashCONNECT administrative tasks and the location in which to access them:

FlashCONNECT Maintenance Web Pages

- Setting up and configuring server pools.
- Setting up groups and users.
- Performing maintenance functions for users, groups and applications.
- Adding, editing, deleting or assigning groups and/or users to applications.
- Running reports about sessions, concurrent use, redirection use, rotating content, line status and content.

D3 TCL

- Starting and stopping FlashCONNECT.
- Checking FlashCONNECT status.
- Configuring FlashCONNECT server pool connections (or from the Server Pool Maintenance Web page).
- Performing maintenance functions for users, groups and applications (or from the User Groups Web page).
- Enabling FlashCONNECT on another account (or from the Applications Web page).
- Viewing FlashCONNECT revision history and reports from the TCL window.

FlashCONNECT Socket Server Manager

- Setting up, starting, stopping, and maintaining socket servers.

Windows

- Changing default settings from the Configure FlashCONNECT program (accessed from the FlashCONNECT program group).
- Starting and Stopping FlashCONNECT from the Services dialog box.

UNIX

- Changing default settings by editing the flashconnect.conf configuration file.

FlashCONNECT administration is generally the same for both the Windows and UNIX platforms. FlashCONNECT administration specific to the Windows platform is presented in [Customizing the Local Environment](#).

Information about setting up FlashCONNECT components is located in:

- [Maintaining Applications, Groups, and Users](#)
- [Administering Server Pools](#)
- [Accessing Administration Reports](#)
- [Administering the FlashCONNECT Socket Server Manager](#)
- [Understanding FlashCONNECT Java Components](#)
- [Configuring, Starting and Stopping FlashCONNECT Components](#)
- [Customizing the FlashCONNECT Server and Web Server](#)
- [Administering Log and JOBS Files](#)
- [Using Character Translation](#)

w3Users, w3Apps, and w3Groups Files

For greater control, the FlashCONNECT Administrator can choose to use the w3Users, w3Groups, and w3Apps files to control access to FlashCONNECT dynamic applications. Using these files, the FlashCONNECT Administrator can choose which users or user groups have access to specific applications. (Use either TCL or the FlashCONNECT Admin Maintenance Menu Web pages.)

NOTE— Review user-IDs, applications, and group definitions to minimize unauthorized access to the system.

Authentication Process

The FlashCONNECT authentication process uses three files:

- w3Users
- w3Groups
- w3Apps

The authentication process differs for static and dynamic applications:

- **Static.** FlashCONNECT authentication is not necessary since static Web pages do *not* interact with the database. FlashCONNECT static applications do not appear in the w3Apps file. Only the standard Web server based authentication, as explained above, is available.

- **Dynamic.** Dynamic Web pages that do interact with the database may require group level and/or user level authentication. When a request comes in to a FlashCONNECT server to run an application, these checks are made:

Step	Check	Yes	No
1	Does the application specified in the w3Exec variable exist in w3Apps file?	Step 2	Error display
2	Is this an anonymous application?	Step 8	Step 3
3	Is the user name valid?	Step 4	Logon display
4	Is the user's password correct?	Step 5	Logon display
5	Does the user field of w3Apps record contain "all"?	Step 8	Step 6
6	Is the user allowed to use the application?	Step 8	Step 7
7	Does one of the groups this user belongs to match one of the groups that can run this application?	Step 8	Logon display
8	Is there an account name and password specified in the w3Apps item?	Log to that account. Step 9	Step 9
9	Run the application.		

Maintaining Applications, Groups, and Users

Use the FlashCONNECT Maintenance Web pages to set up, delete and maintain FlashCONNECT users and groups, and FlashCONNECT application privileges.

After users and groups are set up, assign them access to dynamic applications in the w3Apps file.

NOTE— This topic describes performing various administrative tasks from the FlashCONNECT File Maintenance Web page. These menus and maintenance screens are also accessible from the www account. For example:

1. Log to the www account from the D3 or mvBase database (if it is not the current account).
2. Type **m** to display the FlashCONNECT Support & Maintenance menu (if the menu is not currently displayed).

The FlashCONNECT Support & Maintenance menu displays.

3. Select option 4 (Maintain FlashCONNECT Files).

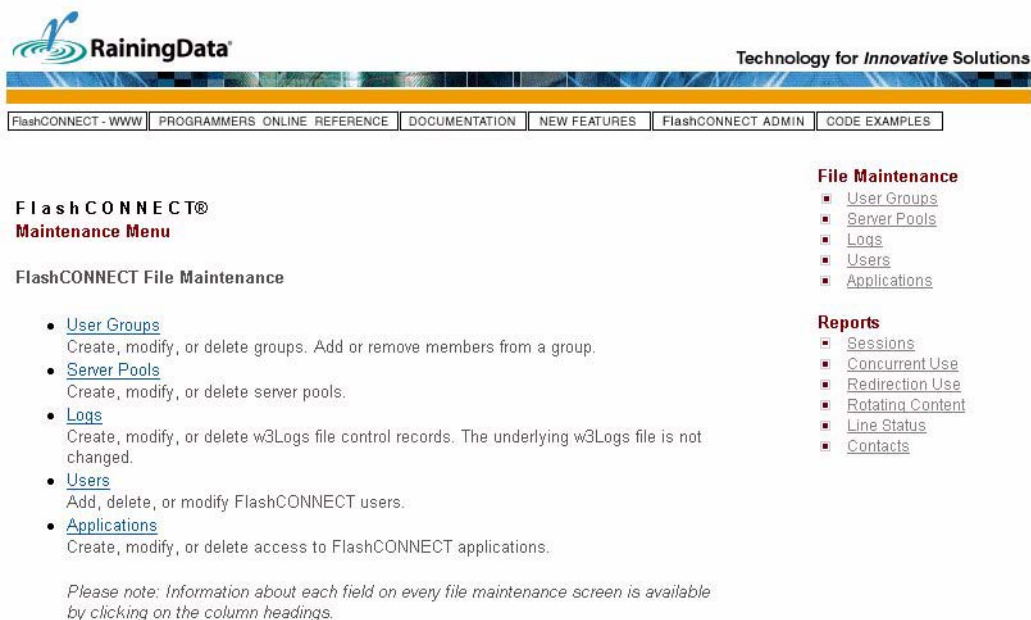
The FlashCONNECT File Maintenance Menu displays.

Logging On to FlashCONNECT Maintenance Pages

Access the FlashCONNECT Maintenance Web pages via your browser using the host name or URL address assigned to your Web server. For example:

http://*yourwebhost*/w3library/w3admin.htm

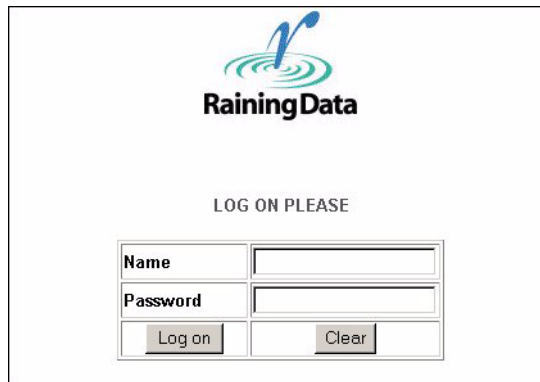
The FlashCONNECT Maintenance Menu displays.



To log on to a FlashCONNECT Maintenance Page:

1. Select the FlashCONNECT Maintenance page to log to (for example, User Groups or Server Pools).

The Log On screen displays.

The image shows a web-based login interface for RainingData. At the top center is the RainingData logo, which consists of a stylized blue and green graphic above the text "RainingData". Below the logo, the text "LOG ON PLEASE" is centered. Underneath this text is a form with two input fields: "Name" and "Password". Below these fields are two buttons: "Log on" and "Clear". The entire form is enclosed in a thin black border.

2. Enter the name and password (as stored in the W3USERS file).

NOTE— If you do not want the Log On page to display, update the w3Apps file entry for the w3Admin application to enable the Anonymous option.

The specified page displays:

NOTE— Screen information is maintained should a timeout occur. So, if you have entered any data prior to the timeout, that information is processed and updated after logging on again.

Adding FlashCONNECT Groups

Use the FlashCONNECT Group Maintenance Web page to add, edit or delete groups that are allowed to access dynamic applications.

Members of groups are restricted to the applications assigned to that group. However, if a member belongs to more than one group, that member may have access to other applications as well.

TIP— If all of your FlashCONNECT users can also log on to the D3 system, then the dm,users, file can be used as the w3Users file. See [w3Users, w3Apps, and w3Groups Files](#) for more information.

To add FlashCONNECT groups from the FlashCONNECT Group Maintenance Web page:

1. Click User Groups from the File Maintenance menu.

The FlashCONNECT User Groups Web page displays.

FlashCONNECT Group Maintenance

The FlashCONNECT User Group Maintenance screen allows creation, deletion, or modification of User Groups. User Groups are designed to simplify granting or denying access to FlashCONNECT applications. Several users can be placed in a group, then permission to use an application is granted to the group as a whole.

Information about each field is available by clicking on the underlined column headings. Detailed information about each feature is available in the manuals.

<u>Delete Group</u>	<u>Name*</u>	<u>Members</u>	<u>Non-Members</u>
<input type="checkbox"/>	Development	Betsy jack	
<input type="checkbox"/>	Applications	Betsy jack	
<input type="checkbox"/>	Administration	Betsy jack	
New Group	<input type="text"/>		Betsy jack

* = Required

2. Type a name for the group in the text box located in the Name column.
3. Click Submit.

Deleting FlashCONNECT Groups

Deleting groups permanently removes the group and its members from accessing any applications.

To delete groups:

- From the FlashCONNECT Group Maintenance Web page, select the check box of the group you want to delete, then click Submit.

Adding Users to Groups

Every user name in the w3Users file is in either the Members or Non-Members list box. However, you can manually add or delete users from the FlashCONNECT User Maintenance Web page or by editing the w3Users file.

NOTE— If the users are already available in the Members and Non-Members list, you may add users from the FlashCONNECT Group Maintenance Web page.

To add FlashCONNECT users to groups:

- Click Users from the File Maintenance menu.

The FlashCONNECT User Maintenance Web page displays.

FlashCONNECT User Maintenance

The FlashCONNECT User Maintenance screen allows creation, deletion, or modification of FlashCONNECT Users. A FlashCONNECT User is only needed if applications requiring logging on are in use.

Information about each field is available by clicking on the underlined column headings. Detailed information about each feature is available in the manuals.

User No.	Delete User	User ID*	User Name	Password	Member	Non-Member
1	<input type="checkbox"/>	Adam	Adam Administrator		Development	Applications Administration
2	<input type="checkbox"/>	Guest	Guest Account		Administration Applications	
3	<input type="checkbox"/>	Mary	Mary McDonald		Applications	Development Administration
4	<input type="checkbox"/>	Oliver	Oliver McBean		Applications	Development Administration
5	<input type="checkbox"/>	www	www			Development Applications
Add User						Development Applications Administration

- Type a unique ID for the new user in the User ID column.

3. Type the user's name in the User Name column.
4. (Optional.) Type a password for the user in the Password column.

The password may include a maximum of 28 alphanumeric characters.

5. Select the group or groups to which you want to add the member from the Non-Member list, then click Submit.

The groups for which the user is a member display in the Member list. The user name also displays in the Members list on the FlashCONNECT Group Maintenance Web page.

NOTE— Select multiple user names from the Non-Member list by holding down the Ctrl key while making a name selection.

Removing and Adding Users from Membership Lists

You can add or remove a user from any group (without deleting the user) at any time by moving the user to and from the Member and Non-Member lists.

To add and remove users from group list:

- From the FlashCONNECT Group Maintenance Web page, select the user name from either the Member list or the Non-Member list, then click Submit.

If the user name is displayed in the Members list, then that user is considered part of the group. If the user name is displayed in the Non-Member list, that user is no longer a member of the group.

Deleting FlashCONNECT Users

Users may be deleted from any group at any time. Removing users from the Member list, deletes them from the group.

To delete FlashCONNECT users from groups:

1. From the FlashCONNECT User Maintenance Web page, locate the Group and select the check box located in the Member column.
2. Click Submit.

The user name file is deleted from the groups and applications for which it was a member.

Assigning Users and Groups to Dynamic Applications

You can assign users and groups to dynamic applications through the w3Apps file or from the FlashCONNECT Applications Selection Web page.

If you use the FlashCONNECT Applications Selection Web page, a list of your applications is available from the drop-down list box.

FlashCONNECT Applications Selection

The FlashCONNECT Application Selection screen allows selection of a FlashCONNECT Application definition entry to delete or modify or to request a new Application entry be created. If the "edit" button is pressed, the Application definition selected in the pull down list is edited. If the "new" button is pressed, a blank Application definition is provided. If the "delete" button is pressed, the Application definition selected in the pull down list is deleted. **By default, the first entry in the pull down list is selected.**

act*22090
act*22090
act*22091
act*22411
act*23235
act*23388
act*24322
act*24326
act24843
applet1
applet2
boo

Edit Delete New Reset

To assign users and groups to new or existing applications, or to edit application information using the FlashCONNECT Applications Selection Web page:

1. Click Applications from the FlashCONNECT File Maintenance menu.

The FlashCONNECT Applications Selection Web page displays.

2. Select an application from the drop-down list box, then click Edit.

The FlashCONNECT Applications Maintenance Web page displays.

FlashCONNECT Applications Maintenance

The FlashCONNECT Application Maintenance screen allows creation or modification of FlashCONNECT application definition entries. The criteria for access each FlashCONNECT application is set in this screen.

Please exit from this screen by using the Cancel, Add, or Update button.

Information about each field is available by clicking on the underlined column headings. Detailed information about each feature is available in the manuals.

<u>Application*</u>	<u>Description</u>	<u>Users</u>	<u>Program</u>		
act*22090	Test action item 22092	Members jack Non-Members betsy			
<u>Account</u>	<u>Password</u>	<u>Options</u>	<u>Groups</u> <u>Timeout</u> <u>Category</u>		
		<input type="checkbox"/> Use SSL <input checked="" type="checkbox"/> Anonymous <input type="checkbox"/> Profile <input type="checkbox"/> Debug Mode	Members Development Non-Members Applications		Test

* = Required

3. (Optional.) Type any additional application information in the Description text box.
4. Select the users you want to access this application from the Non-Members list box.
5. (Optional.) Type the application name in the Program text box if the name differs from the application name.
6. (Optional.) Type the account name that the application should run from if it differs from the www account.
7. (Optional.) Type the password for the account if it is required for that account.
8. (Optional.) Enable any of the following options:

Option	Description
Use SSL	Enables a secure socket layer between the browser and Web server.
Profile	Remembers the profile ID and enables users to logon without using a password (provided the password did not change since profiling was enabled).

Option	Description
Anonymous	Enables users access to applications without entering their user ID or password.
Debug Mode	Enables debugging capabilities.

9. Select the names of the groups that you want to access this application from the Non-Members list box.
10. (Optional.) Type the number of seconds in the Timeout text box to indicate the amount of time to the user name and password is remembered if Profile is selected. Set Timeout to 0 or null to disable this feature.
11. (Optional.) Type the name of the category to which this application belongs.

NOTE— The category field located on the FlashCONNECT Applications Web page is a new feature that allows convenient grouping of applications. For example, group your debugging, programming or accounting applications together in one category and then locate them by typing **select w3Apps by category**. This is especially helpful if you intend to use mvDesigner since your applications are listed by category in the tree view of the Enterprise browser.

12. Click Update.

Deleting Applications

Applications may be deleted from the w3Apps file so that they are no longer available from the Application list.

To delete applications using the FlashCONNECT Applications Selection Web page:

1. Select the application you want to delete from the drop-down list box.
2. Click Delete.

Administering Server Pools

FlashCONNECT's server pooling extends the concept of connection pooling. Connection pooling enables applications to reuse a connection from a pool of connections. Once a connection is created and placed in the pool, an application can use a connection within the same pool without performing the complete connection process. Connections (when released by an application) remain available, waiting for the next occasion they are needed.

Using Server Pools

Server pooling extends connection pooling capabilities by allowing the naming and grouping of one or more FlashCONNECT connections. Server pools can connect one or more database servers to one or more Web servers. Server pools enable the distribution of an application across several systems, and allow load balancing and tighter control of resource allocation. Setting up, configuring and using server pools are optional.

- The URL specifies which system should process a request and which server pool to use.
- FlashCONNECT assigns the connection to any database server which supports that server pool using the value assigned to the `w3ServerPool` variable.

Starting Server Pools in Groups or Individual Lines Manually

To start server pools in groups:

When starting FlashCONNECT with `WWW-MANAGER`, `WWW-MANAGER` uses the `WWW-POOL` command to enable each configured server pool and starts the `w3Monitor`. The `w3Monitor` uses the `WWW-START` command to start the `WWW-IRUN` phantom processes, which service the FlashCONNECT enabled Web pages.

To start individual lines:

When both the Web server and the FlashCONNECT service are properly configured and started, start the FlashCONNECT servers.

1. Log to the `www` account.
2. Use the `WWW-START` command to start the servers. For example:

```
WWW-START NUMSERVERS HOST PORT [POOLNAME]
```

where

<i>NUMSERVERS</i>	Number of new FlashCONNECT servers to start.
<i>HOST</i>	TCP/IP address or host name that the FlashCONNECT service is running on.
<i>PORT</i>	TCP/IP port number that the FlashCONNECT service is listening on. Port 1505 is the default port number.

POOLNAME Server pool name of this set of FlashCONNECT servers. The name is case-sensitive. If left blank, these servers are added to the default pool.

For example, to start ten connections in the GAME server pool on www.rainingdatagames.com (assuming that www.rainingdatagames.com has a Web server and FlashCONNECT service running on it), use this command:

```
WWW-START 10 WWW.RAININGDATAGAMES.COM 1505 GAME
```

Stopping Server Pools in Groups or Individual Lines Manually

To stop server pools in groups:

When stopping FlashCONNECT with `WWW-MANAGER`, `WWW-MANAGER` stops the `w3Monitor` and then uses `WWW-STOP` to shut down each configured server pool. Occasionally, it is necessary to manually start and stop both the `w3Monitor` and FlashCONNECT connections. This process stops the `w3Monitor` from the configuration file.

To stop individual lines or server pools:

Use the `WWW-STOP` command from the `www` account to stop FlashCONNECT connections.

```
WWW-STOP [SERVER-POOL | DATABASE LINE] { (OPTIONS) }
```

where

<i>SERVER-POOL</i>	Name of server pool to stop.	
<i>DATABASE LINE</i>	Database line to stop that is running FlashCONNECT (Port).	
<i>OPTIONS</i>		
	F	Forces the database server to disconnect and closes sockets when <code>WWW-STOP</code> fails, (possibly because all FlashCONNECT lines are in use). Only stopped lines are impacted.
	Q	Stops all lines in the server pool, stops those connected to this database.

NOTE— If `WWW-STOP` is issued with neither a PORT or a POOLNAME specified, no action is taken

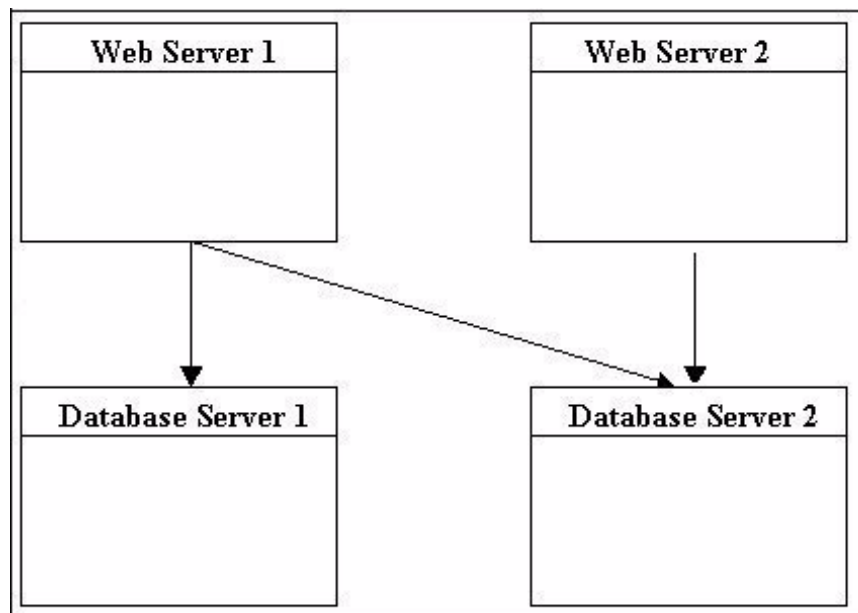
Examples:

- To stop all GAME servers, use the command:
`WWW-STOP GAME`
- To stop only the server running on database line number 98, use the command:
`WWW-STOP 98`
- To stop all entries in the default server pool on this database server, use the command:
`WWW-STOP DEFAULT`

Connecting Database Server Pools

FlashCONNECT allows one or more database servers to connect to one or more FlashCONNECT enabled Web servers. Using server pools other than the default server pool is optional. However, if other server pools are used, the default server pool does not need to be used.

In this example, requests to Web Server 1 can be directed to Database Server 1 or Database Server 2, while requests to Web Server 2 can only be directed to Database Server 2.



These pools enable the distribution of an application across several systems.

- The system administrator specifies which server pool a system supports using the `WWW-START` command or via server pools configuration.
- The programmer specifies which server pool is used to service a request by setting the `w3ServerPool` variable.
- FlashCONNECT assigns the connection to any database server that supports the requested server pool. If no server pool is specified, the default server pool is used.

WARNING— • Server Pool names are case sensitive.

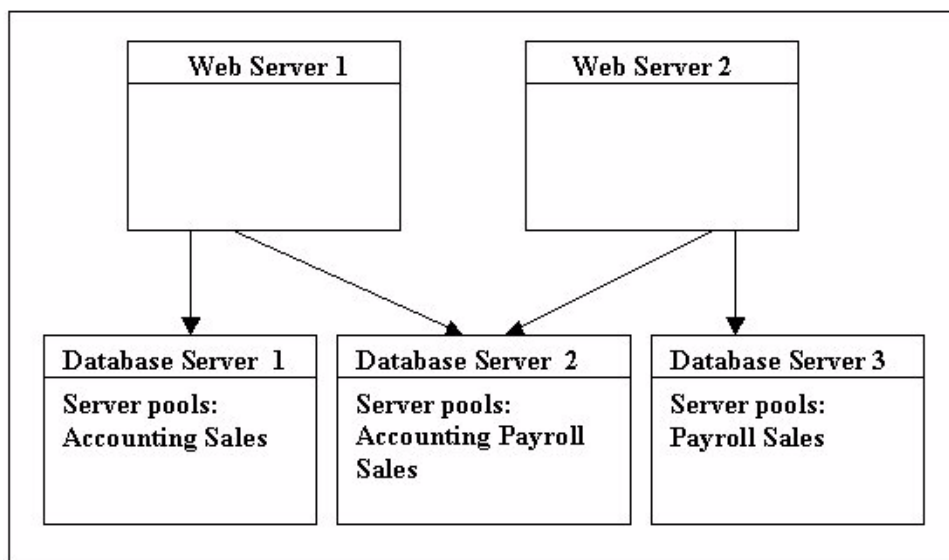
- Also, ensure that the server pool name is not an ADI or data level in the `w3lines` file.
-

CAUTION—The database server pool name can be a maximum of 15 alphanumeric characters and must include at least one alpha character in the name.

NOTE— `w3ServerPool` is a synonym for `w3HostName`. In this document, all `w3HostName` references have been changed to `w3ServerPool`.

For server pool debugging assistance, see [Debugging Server Pools](#).

The Web servers are directed to different database servers as displayed in this example:



The following example shows which database servers are available to each Web server when a request to the server pool is made.

Requested Pool	Accounting	Payroll	Sales
Web Server 1	Database Server 1 or 2	Database Server 2	Database Server 1 or 2
Web Server 2	Database Server 2	Database Server 2 or 3	Database Server 2 or 3

Specifying the Relationship

Specify which system should process a request and which server pool to use in processing that request. Configure server pools using the FlashCONNECT Server Pool Maintenance Web page.

Then use the `WWW-MANAGER` command to implement the configuration at startup. Create a macro containing all of the desired commands, and call that macro from within the user-coldstart macro. The macro should log to the `www` account, then process commands, and then log back to the `dm` account. The following `w3Config` entries are used to establish the connection for the previous examples. Note that the `Min`, `Max`, `MaxFree`, `MinFree`, `Freq`, `Port`, and `HB Period` are arbitrarily set for this example.

Database Server 1:

Server pool: Accounting	Min	Max	MaxFree	MinFree	Freq	Port	HB Period
WEBSERVER1	1	3	2	1	60	1505	120
Server pool: Sales							
WEBSERVER1	1	3	2	1	60	1505	120

Database Server 2:

Server pool: Accounting	Min	Max	MaxFree	MinFree	Freq	Port	HB Period
WEBSERVER2	1	3	2	1	60	1505	120
WEBSERVER1	1	3	2	1	60	1505	120
Server pool: Payroll							

Server pool: Accounting	Min	Max	MaxFree	MinFree	Freq	Port	HB Period
WEBSERVER1	1	3	2	1	60	1505	120
WEBSERVER2	1	3	2	1	60	1505	120
Server pool: Sales							
WEBSERVER1	1	3	2	1	60	1505	120
WEBSERVER2	1	3	2	1	60	1505	120

Database Server 3:

Server pool: Payroll	Min	Max	MaxFree	MinFree	Freq	Port	HB Period
WEBSERVER1	1	3	2	1	60	1505	120
Server pool: Sales							
WEBSERVER2	1	3	2	1	60	1505	120

The following commands manually start the same server pools as the above example:

Database Server 1:

```
WWW-START 1 WEBSERVER1 1505 ACCOUNTING
WWW-START 1 WEBSERVER1 1505 SALES
```

Database Server 2:

```
WWW-START 1 WEBSERVER1 1505 ACCOUNTING
WWW-START 1 WEBSERVER1 1505 PAYROLL
WWW-START 1 WEBSERVER1 1505 SALES
WWW-START 1 WEBSERVER2 1505 ACCOUNTING
WWW-START 1 WEBSERVER2 1505 PAYROLL
WWW-START 1 WEBSERVER2 1505 SALES
```

Database Server 3:

```
WWW-START 1 WEBSERVER2 1505 PAYROLL
WWW-START 1 WEBSERVER2 1505 SALES
```

Assigning the Connection

FlashCONNECT assigns the connection to any database server that supports that server pool. Use the FlashCONNECT variable `w3ServerPool` to specify which pool the

FlashCONNECT enabled Web server should use. In the following examples, *Apps* represents an item in the w3Apps file.

Using the information in the Web Server 1 row in the table above:

- to specify Web Server 1 for the Accounting server pool to Database Server 1 or 2, use the URL:

`http://WebServer1/cgi-bin/fcisapi.dll?w3exec=Apps &w3ServerPool=Accounting`

- to specify Web Server 1 for the Payroll server pool to Database Server 2, use the URL:

`http://WebServer1/cgi-bin/fcisapi.dll?w3exec=Apps &w3ServerPool=Payroll`

Using the information in the Web Server 2 row in the table above:

- to specify Web Server 2 for the Accounting server pool to Database Server 2, use the URL:

`http://WebServer2/cgi-bin/fcisapi.dll?w3exec=Apps &w3ServerPool=Accounting`

- to specify Web Server 2 for the Payroll server pool to Database Server 2 or 3, use the URL:

`http://WebServer2/cgi-bin/fcisapi.dll?w3exec=Apps &w3ServerPool=Payroll`

Automatic Load Balancing

When providing the same service on multiple database servers, the load can be automatically balanced between systems using server pools and w3Monitor. This is possible because:

- connections from multiple database servers can be assigned to the same server pool on one or more Web servers.
- FCService uses server pool entries to any database server supporting that server pool.
- w3Monitor creates and destroys connections as needed.

See [w3Monitor Application](#) in the FlashCONNECT Programmer's Guide for additional information.

To implement automatic load balancing:

1. Specify the relationship (which system should process a request and which server pool to use in processing that request). See [Specifying the Relationship](#).

2. Assign the connection (to any database server which supports that server pool). See [Assigning the Connection](#).
3. Set up w3Monitor on each database server to maintain a minimum of one or more free connections to the Web server.

How it Works:

For example, there are two database servers (A and B) supporting one Web server. Two connections to the default server pool are made from each database server to the Web server (Table 2). w3Monitor is set up on each database server to maintain a minimum of one free connection to the Web server.

Table 1: Two database Servers Support One Web Server

Web Server Connection Number	Host	Assigned
1.	Database A	No
2.	Database A	No
3.	Database B	No
4.	Database B	No

The first two user requests are assigned to connections 1 and 2, both to Database A. At this time, the w3Monitor running on Database A automatically adds one more connection (number 5), as reflected in Table 3.

Table 2: w3Monitor Automatically Adds a Connection

Web Server Connection Number	Host	Assigned
1.	Database A	Yes
2.	Database A	Yes
3.	Database B	No
4.	Database B	No
5.	Database A	No

The third and fourth user requests are assigned to connections 3 and 4, both to Database B. At this time, the w3Monitor running on Database B automatically adds one more connection (number 6) as shown in Table 4.

Table 3: w3Monitor Automatically Adds another Connection

Web Server Connection Number	Host	Assigned
------------------------------	------	----------

1.	Database A	Yes
2.	Database A	Yes
3.	Database B	Yes
4.	Database B	Yes
5.	Database A	No
6.	Database B	No

From this point forward, as additional connections are requested, they are alternately assigned to each database server. w3Monitor creates additional connections as needed until the configuration limits are reached, thus balancing the load.

As connections are dropped, they are reassigned in connection number order. For example, if connection 1 and 3 are both available, connection 1 is used before connection 3 is used. At the same time, depending on how the w3Monitor is configured, as the load is shed from one database server, connections are dropped to help maintain load balance as the load diminishes.

NOTE— When additional connections are needed, w3Monitor may create more than one connection at a time, slightly changing the balancing pattern from the pattern shown here.

Controlling Server Pools and w3Monitor Manually

Usually `WWW-MANAGER` is used to start and stop both the server pools and w3Monitor. However, if manual control is needed, or more visibility of w3Monitor's action is needed, both the server pools and w3Monitor can be manually controlled.

If more visibility of w3Monitor's activities is needed, w3Monitor can also run from a terminal (for example, by logging to the `WWW` account from a terminal and typing `W3MONITOR -START`).

w3Monitor only manages enabled server pools. The `WWW-MANAGER`, `WWW-POOL` and `WWW-START` commands are used to start a server pool, while `WWW-STOP` and `WWW-MANAGER` are used to stop a server pool.

NOTE— For information about configuring server pools, see [Configuring FlashCONNECT Server Pools](#) in the FlashCONNECT Installation Guide.

Accessing Administration Reports

The FlashCONNECT Maintenance Menu Pages provide access to several administration reports:

- [Concurrent FlashCONNECT Use Report](#)
- [FlashCONNECT Sessions Report](#)
- [FlashCONNECT Contacts Report](#)
- [FlashCONNECT Redirect Report](#)
- [FlashCONNECT Rotating Content Report](#)
- [FlashCONNECT Line Status Report](#)

FlashCONNECT report descriptions and sample reports are available here and online.

For information about logging on to the FlashCONNECT Administration Web pages, see [Logging On to FlashCONNECT Maintenance Pages](#).

View the report sources in the www account located in the BP file.

Concurrent FlashCONNECT Use Report

The Concurrent FlashCONNECT Use report indicates the number of FlashCONNECT ports being used during various periods. Use this report to see the volume of traffic and to determine if more connections are needed.

Periods with no usage are not shown. Each FlashCONNECT port is counted only once, so the information represents an estimate of the number of concurrent FlashCONNECT users.

Concurrent FlashCONNECT Use		
Reporting period 300 seconds		
Date	Time	# Users
03/15/01	08:54:53	1
03/15/01	08:59:53	1
03/15/01	09:04:53	1
03/15/01	09:09:53	1
03/15/01	09:14:53	1
03/15/01	09:19:53	3
03/15/01	09:24:53	2
03/15/01	09:29:53	1
03/15/01	09:54:53	1
03/15/01	10:09:53	1
03/15/01	10:14:53	1
03/15/01	10:59:53	1
03/15/01	14:24:53	1
03/16/01	09:29:53	1

FlashCONNECT Sessions Report

The FlashCONNECT Sessions Report provides information about each completed FlashCONNECT session. Use this report to determine who is running what applications.

FlashCONNECT Sessions Report							
Port	Date	Time	Duration	w3ProfileId	Client Address	Programs	User
25	03/14/01	16:20:45	00:00:00	1212758540285202	172.16.129.86	w3Admin Unstarted	-
25	03/15/01	08:54:53	00:00:00	121283209320867	172.16.128.64	java1	-
25	03/15/01	08:55:59	00:00:01	120734354615843	172.16.250.3	w3Admin	-
25	03/15/01	09:00:12	00:00:00	120734354615843	172.16.250.3	w3Admin	-
25	03/15/01	09:04:02	00:00:00	120734354615843	172.16.250.3	w3Admin	-
25	03/15/01	09:07:01	00:00:01	120734354615843	172.16.250.3	w3Sessions.html	-
25	03/15/01	09:07:06	00:00:01	120734354615843	172.16.250.3	w3Concurrent.html	-
25	03/15/01	09:07:10	00:00:00	120734354615843	172.16.250.3	w3Redirect.html	-
25	03/15/01	09:07:13	00:00:00	120734354615843	172.16.250.3	w3Rotator.html	-
25	03/15/01	09:07:14	00:00:01	120734354615843	172.16.250.3	w3Lines.html	-
25	03/15/01	09:07:19	00:00:01	120734354615843	172.16.250.3	w3Contacts.html	-
25	03/15/01	09:07:21	00:00:01	120734354615843	172.16.250.3	w3Lines.html	-
25	03/15/01	09:07:24	00:00:00	120734354615843	172.16.250.3	w3Rotator.html	-
25	03/15/01	09:07:25	00:00:00	120734354615843	172.16.250.3	w3Redirect.html	-

The following information is reported for each completed session:

- FlashCONNECT port used.
- Session starting date.
- Session starting time.
- Session duration including the hours, minutes and seconds that the process was run.
- FlashCONNECT Profile ID.
- Client IP address.
- Programs or applications accessed.
- FlashCONNECT username (if applicable).

FlashCONNECT Contacts Report

The FlashCONNECT Contacts Report shows the volume of FlashCONNECT traffic, providing the raw number of active FlashCONNECT sessions (or hits) during a time period.

FlashCONNECT Contacts Report		
Reporting period 300 seconds		
Date	Time	# Sessions
03/15/01	08:54:53	1
03/15/01	08:59:53	1
03/15/01	09:04:53	2
03/15/01	09:09:53	10
03/15/01	09:14:53	6
03/15/01	09:19:53	7
03/15/01	09:24:53	2
03/15/01	09:54:53	1
03/15/01	10:09:53	7
03/15/01	10:14:53	1
03/15/01	10:59:53	6
03/15/01	14:24:53	2
03/16/01	09:29:53	2
03/16/01	09:34:53	1
03/16/01	09:44:53	1

The date and time, as well as the number of FlashCONNECT sessions, is listed.

FlashCONNECT Redirect Report

The FlashCONNECT Redirect Report lists the number of FlashCONNECT click-throughs to other sites. Use this report to monitor the click-throughs that have occurred on your site.

FlashCONNECT Redirect Report						
Date	Time	Port	Source	Destination	Client Address	w3ProfileId

The following information is reported:

- Click-through date and time.
- FlashCONNECT port used.
- Web page containing the link or the page that the user navigated from.
- Destination or the page the user navigated to.
- Client IP address.
- FlashCONNECT Profile ID.

FlashCONNECT Rotating Content Report

The FlashCONNECT Rotating Content Report details the use of rotating content. Since rotating content can be a revenue source, you may want to use this report to monitor it.

FlashCONNECT Rotating Content Report					
Date	Time	Port	Client Address	w3ProfileId	Value Used

The following information is reported:

- Date and time used.
- FlashCONNECT port used.
- Client IP address.
- FlashCONNECT Profile ID.
- Value used or the name of the object being used.

For more information, see the online code examples in the [FlashCONNECT Programmer's Online Reference](#).

FlashCONNECT Line Status Report

The FlashCONNECT Line Status Report details the current connection status. Use this report to monitor the line status.

FlashCONNECT Line Status Report											
Line	Pool	Host	Port	Status	Since	Socket	Application	User	Index	Up/Down	Options
25	default	localhost	1505	Processing	09:20:52 15 Mar 2001	660	-	-	5	up	-
26	default	localhost	1505	Input	09:20:48 15 Mar 2001	576	-	-	7	up	-
27	default	localhost	1505	Processing	09:49:10 16 Mar 2001	648	w3Lines.html	-	6	up	-
28	default	localhost	1505	Input	11:25:52 14 Mar 2001	824	-	-	8	up	-

The following information is reported:

- FlashCONNECT line number.
- Server Pool name.
- Server host name.
- FlashCONNECT Listening Server port.
- Status from the w3lines file.
- Start date and time.
- Socket or the TCP/IP port on the D3 server.
- Application being run.
- User name, if any.
- Index or incremental number on the Web server.
- Up/Down indicates whether the connection is up or down.
- Options defined in the W3Apps file for FlashCONNECT options.

Administering the FlashCONNECT Socket Server Manager

FlashCONNECT uses the FlashCONNECT Socket Server Manager to connect to the mvDesigner client and server. This enables Pick BASIC applications to communicate with mvDesigner forms to provide a graphical display of your database queries.

Accessing the FlashCONNECT Socket Server Manager

The FlashCONNECT Socket Server Manager (fcss.jar) installs with FlashCONNECT, provided a Java Virtual Machine is available on your machine. Please see the FlashCONNECT Installation Guide for more information.

Windows:

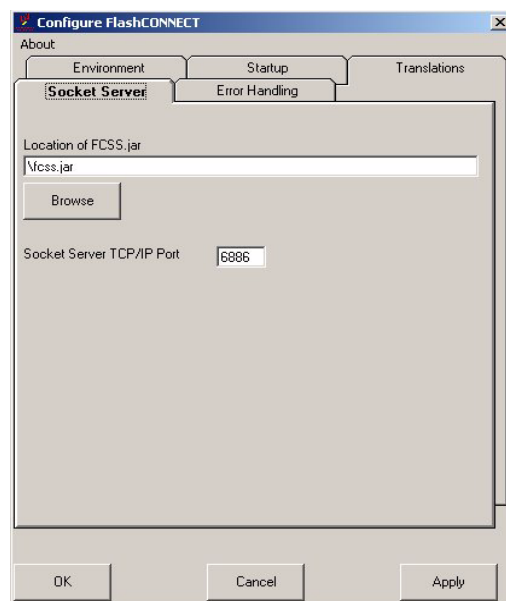
To verify the location of the fcss.jar file:

1. Select Start > Programs > Raining Data > FlashCONNECT > Configure from the Start menu.

The Configure FlashCONNECT dialog box displays.

2. Select the Socket Server tab and make sure that the path to the fcss.jar file is correct.

Click Browse to locate or verify the location of the fcss.jar file.



3. Enter the TCP/IP Port number if it is different from what you are using.

NOTE— The TCP/IP Port number range is 1025-65535.

4. Click Apply.

UNIX:

During installation, enter the path to the directory for the FCSocket server:

Enter the path to the directory for the FCSocketServer script
used to start and stop the FlashCONNECT Java component
[/usr/bin](Q=QUIT):

Starting the FlashCONNECT Socket Server Manager

The FlashCONNECT Socket Server Manager provides a visual interface for starting, stopping and monitoring socket servers.

Windows:

Starting the FlashCONNECT Socket Server Manager from the Control Panel > Services, starts the FlashCONNECT Socket Server Manager and starts the port that you set in the configuration.

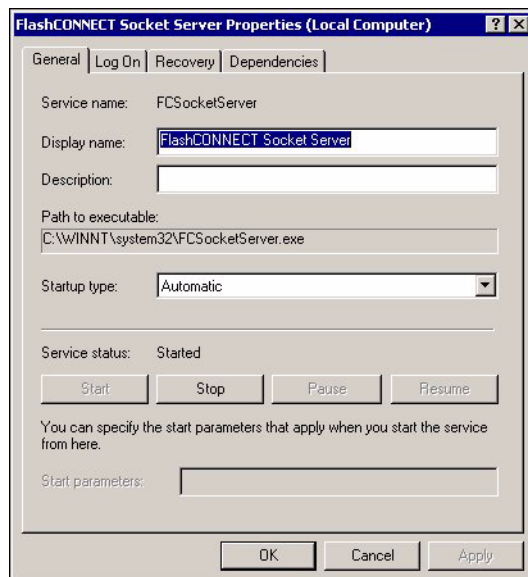
To start the FlashCONNECT Socket Server Manager:

1. Select Start > Settings > Control Panel from the Start menu.
2. Double-click Administrative Tools, then select Services.

The Services dialog box displays.

3. Double-click FlashCONNECT Socket Server.

The FlashCONNECT Socket Server Properties dialog box displays.



4. (Optional) Change any of the following:
 - Display name. Change this to another descriptive name.
 - Description. Add additional information.
 - Startup type. Select Automatic, Manual or Disabled.
 - Start or Stop the Socket Server service.
5. (Optional) Select the Log On tab.
 - a. Select one of the following:
 - Select Local System Account and the Allow Service to Interact with the Desktop check box.

NOTE—Selecting Allow Service to Interact with the Desktop allows you to access the FlashCONNECT Socket Server Manager dialog box which is required if you want to add additional FlashCONNECT Socket Server ports, view log entries or run FlashCONNECT Servlet diagnostics.

- Select This Account and type the account name and password.
- b. Enable or disable the profile.
 1. Select the profile from the Hardware Profile list.
 2. Click Enable or Disable.

Starting the FlashCONNECT Socket Server Manager Manually

This procedure starts the FlashCONNECT Socket Server Manager and the port that you set in the configuration.

Windows:

1. Open the MS-DOS Window.
2. Type:

```
FCSocketServer -start
```

3. Press Enter.

The FlashCONNECT Socket Server Manager dialog box displays.

UNIX:

To run the FlashCONNECT Socket Server Manager graphical user interface, X Windows must be installed. Otherwise, you can only start and stop the FlashCONNECT Socket Servers from these commands:

```
.\fcsocketserver start  
.\fcsocketserver stop
```

Without a GUI, you cannot take advantage of viewing log entries, adding additional FlashCONNECT Socket Server ports or running FlashCONNECT Servlet diagnostic tests.

To start the FlashCONNECT Socket Server Manager:

- At the prompt, type:

```
fcsocketserver start {port {gui}}
```

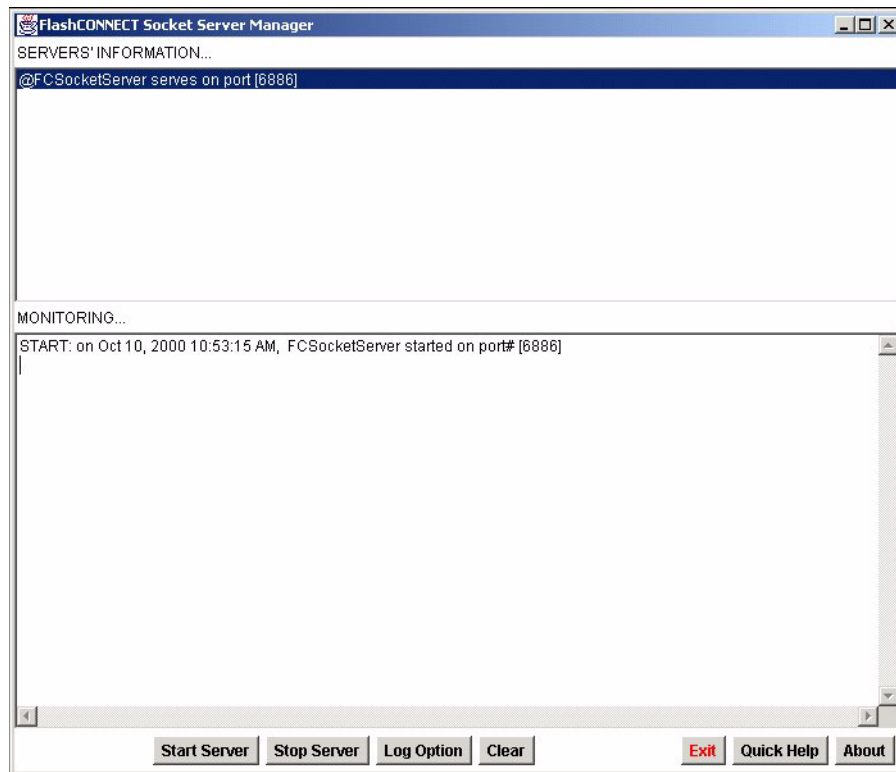
Where *port* indicates the socket server port number (1-65535) and *gui* indicates that X Windows is installed.

The message `Starting FlashCONNECT Socket Server` displays.

The FlashCONNECT Socket Server Manager interface displays.

Using the FlashCONNECT Socket Server Manager

The FlashCONNECT Socket Server Manager dialog box provides server information in the upper pane and monitoring information in the lower pane.



The FlashCONNECT Socket Server Manager allows you to add new socket servers, start and stop socket servers, and monitor socket server events.

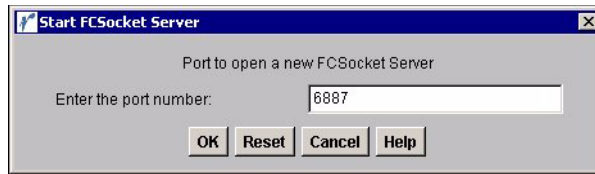
Adding FlashCONNECT Socket Servers

You can add an unlimited number of socket servers depending on the resources available on the computer. However, in most cases, only one socket server is needed.

To add a FlashCONNECT Socket Server:

1. Click Start Server.

The Start FCSocket Server dialog box displays.



2. Type the port number, then click OK.

NOTE— The TCP/IP Port number range is 1025-65535. Click Reset to return to the initial port number.

Stopping FlashCONNECT Socket Servers

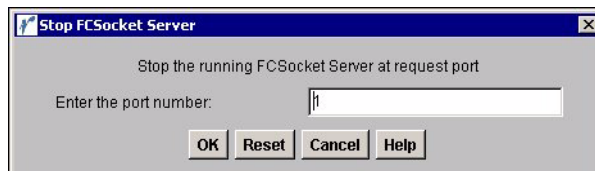
If you have multiple FlashCONNECT socket server ports, you can stop a connection at any time using the FlashCONNECT Socket Server Manager.

To stop a FlashCONNECT Socket Server:

Windows and UNIX:

1. Click Stop Server from the FlashCONNECT Socket Server dialog box.

The Stop FCSocket Server dialog box displays.



2. Type the port number of the socket server that you want to stop.
3. Click OK.

To Stop all FlashCONNECT Socket Servers and close the FlashCONNECT Socket Server Manager:

Windows:

Do one of the following:

- Double-click FlashCONNECT Socket Server from Services (located in the control panel) to open the Properties dialog box. Click Stop.

- Select FlashCONNECT Socket Server from the list. Right-click and select Stop from the pop-up menu.
- Click Exit from the FlashCONNECT Socket Server Manager dialog box.

UNIX:

Do one of the following:

- Click Exit from the FlashCONNECT Socket Server Manager dialog box.
- At the prompt, type:

```
./fcsocketserver stop
```

Terminating FlashCONNECT Socket Server Connections

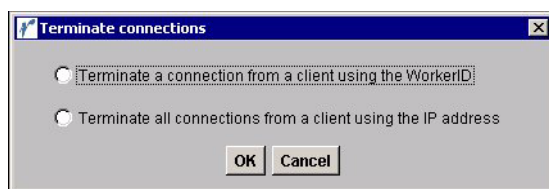
Terminate FlashCONNECT Socket Server connections by:

- terminating the WorkerID of the connected client. The connection is dropped.
- terminating the IP address of the connected client. All client connections using this IP address are terminated.

To terminate FlashCONNECT Socket Server connections:

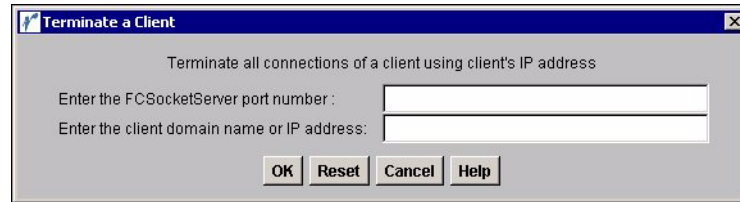
1. Click Terminate connections.

The Terminate connections dialog box displays.



2. Select one of the following options:
 - Terminate a connection from a client using the WorkerID.
The user is disconnected and the connection is closed.
 - Terminate all connections from a client using the IP address.
All client connections using this IP address are terminated.

The Terminate a Client dialog box displays.



3. Type the FlashCONNECT Socket Server port number in the text box.
4. Type either the Client WorkerID or the IP address depending on the type of connection you are terminating.
5. Click OK.

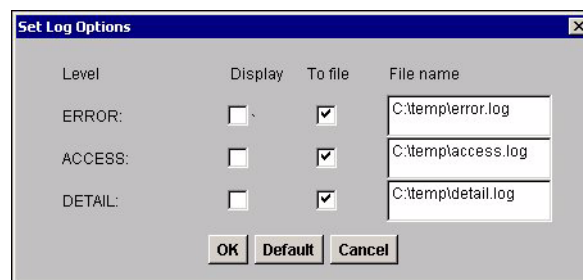
Using the FlashCONNECT Socket Server Log

The FlashCONNECT Socket Server log information can be displayed in either or both the Monitoring pane and log file. The FlashCONNECT Server Log is available if you are using the FlashCONNECT Socket Server Manager graphical user interface.

To select the information to record in the Monitoring pane or Log File:

1. Click Log Option.

The Set Log Options dialog box displays.



2. Select the FlashCONNECT Socket Server options you want to display.

NOTE— • Select the File check boxes to record entries in the log file.
 • Select the Display check boxes to display entries in the Monitoring panel.

The FlashCONNECT Socket Server Log records the following information:

Log Options	Description
Error	Unable to service a request. Internal or protocol error.
Access	Provides logon history and information.
Detail	Displays FlashCONNECT Socket Server and mvDesigner Server messages. Displays each message sent and received by the FlashCONNECT Socket Server Manager. For example, if you attempt to connect to server pool ABC123 and there is no such server pool, or if you connect to FlashCONNECT or D3 and neither is available.

Changing the Log File

The Log file path and directory may be changed.

- Change the directory and path from the Set Log Options dialog box.

Error Codes

Error messages indicate an unsuccessful completion of a request but leave the connection available for future use.

The table below provides a list of error codes and descriptions.

Error Name	Value	Definition
No Connections	001	There are no connections available for the specified server pool.
Invalid Session	002	The w3SessionId is invalid, possibly because the session timed out.
No Such Application	003	Requested application does not exist.
Unauthorized Access	004	The username/password combination is not authorized to run the application.
No Response	005	Application ended without sending a response.
Undecipherable Message	006	Application returned a message that is not in the mvDesigner protocol and was not decipherable as a standard FlashCONNECT error message.

Error Name	Value	Definition
Incorrect w3Exec, w3SessionId	007	Either both a w3Exec and w3SessionId were specified or neither w3Exec or w3SessionId were specified.
Unsupported Feature	008	Feature used is not supported, for example, an unsupported data type.
Maximum Message Length Exceeded	009	Overall message length exceeds the maximum message length.
Closing Page Sent	010	Application used w3ClosingPage to terminate an application which isn't supported in mvDesigner.
Change Account Failed	011	Account password in the w3Apps file is not correct.

Fatal Errors

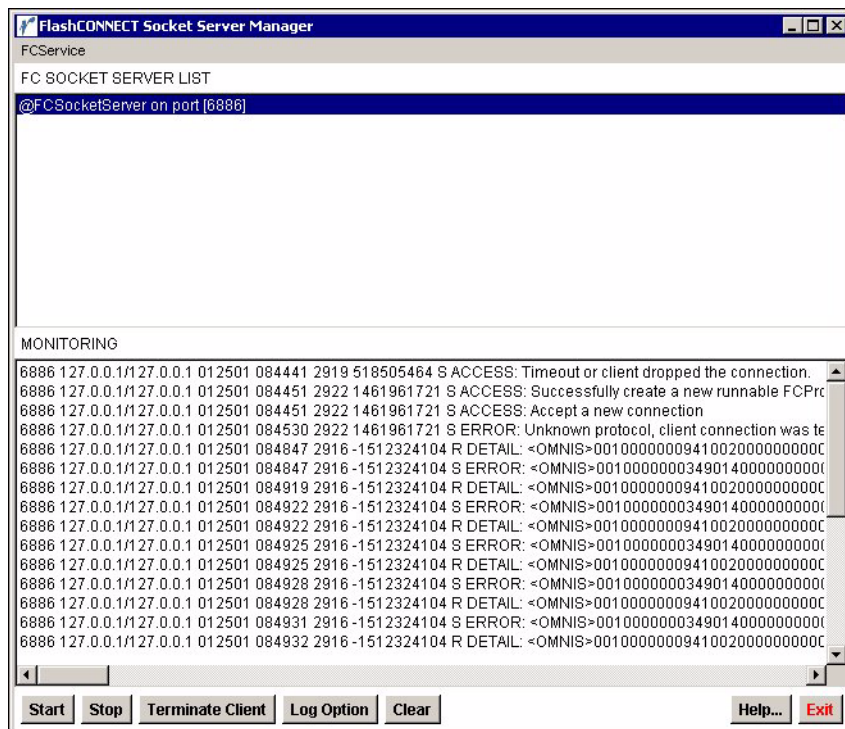
Fatal errors result in a dropped connection. The client must be reconnected to the FC Socket Server. The following fatal errors are generated and displayed (if you select this option) in the FC Socket Server Manager.

- Unknown Protocol. The message does not start with a recognized protocol header.
- Client unexpectedly closes the connection.
- I/O error when sending or receiving a message from the client.

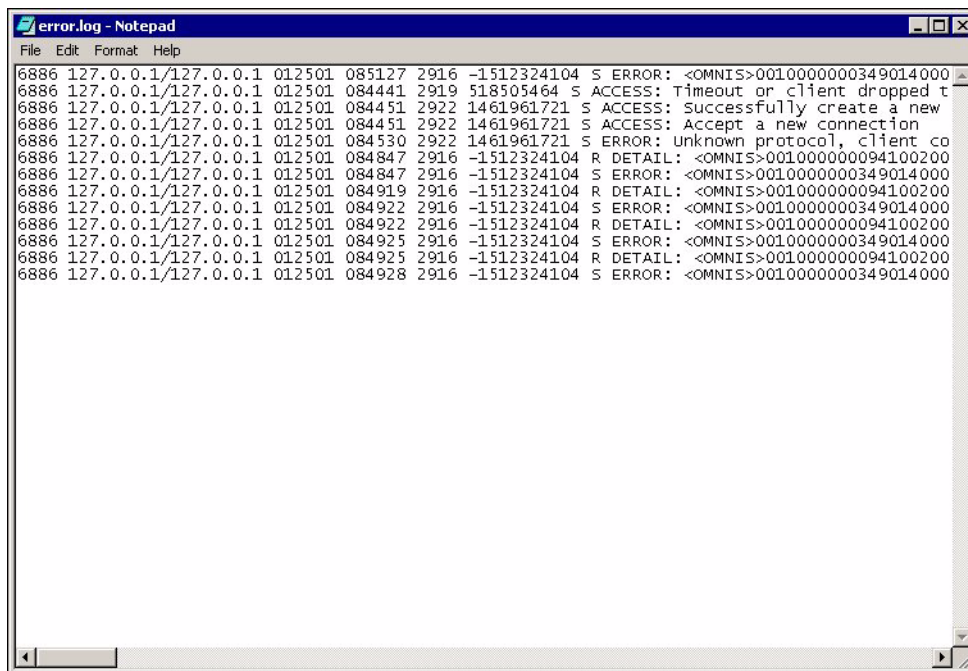
Viewing Monitor Information

The FlashCONNECT Socket Server information may be displayed in the lower pane of the FlashCONNECT Socket Server Manager dialog box and/or from the log file location you designated depending on the Log Options set. See [Using the FlashCONNECT Socket Server Log](#) for information about setting log options.

The following shows the FlashCONNECT Socket Server Manager with error, access and detail log information recorded in the lower panel.



If you elected to have errors logged to a file, it might look something like this:



Understanding Log Information

The following example displays an error code line that was logged:

```
6886 127.0.0.1/127.0.0.1 012501 085127 2916 -1512324104 S
      ERROR:
      <OMNIS>0010000000349014000000000000000030010150000000000
      00290com.pick.br.FCConnectionUnavailableException:
      From FCJavaServer.call(FCRule): The connection to the
      FlashCONNECT failed. Possible causes: 1)FCService is
      not available. OR, 2)D3 Server pool might not start or
      available, or invalid D3 Server pool name. 3)Invalid
      session id value request.
```

The table below explains how to read the code line:

Code Line	Description
6886	FlashCONNECT Socket Server port number.
127.0.0.1/127.0.0.1	Client IP address/Server IP address.
012501	Date the error occurred; (mm:dd:yy).
085127	Time that the error was recorded: (hh:mm:ss).
2916	Client port number.
-1512324104	WorkerID.
S/R	Indicates the direction of the message. S indicates that it was sent from the FlashCONNECT Socket Server. R indicates that FlashCONNECT Socket Server received a message from an external client.
ERROR:	Message type. Can be error, access or detail.

Code Line	Description
<pre><OMNIS>0010000000349014000000 00000000300101500000000000029 0com.pick.br.FCConnectionUnav ailableException: From FCJavaServer.call(FCRule) The connection to the FlashCONNECT failed. Possible causes: 1)FCService is not available. OR, 2)D3 Server pool might not start or available, or invalid D3 Server pool name. 3)Invalid session id value request.</pre>	Error message sent to mvDesigner.

Clearing Monitor Information

From time to time, you may want to delete the information recorded in the Monitoring pane.

To delete the information in the Monitoring pane:

- Select Clear.

Exiting FlashCONNECT Socket Server Manager

Closing the FlashCONNECT Socket Server Manager terminates all FlashCONNECT socket server connections.

1. Click Exit to close the FlashCONNECT Socket Server Manager dialog box.

The message Warning! Do you really want to terminate all FCSocketServers and exit? displays.

2. Click OK.

Understanding FlashCONNECT Java Components

FlashCONNECT contains Java features that enable it to communicate more efficiently with the Web browser. Java components increase the speed of data transfer by enabling better communication between the Web server and FCServlet. The Java components include FCServlet, an FCServlet Diagnostic application, and a Java Native Interface (JNI).

FCServlet

The Java Servlet interface between the Web server and FCService provides important benefits for FlashCONNECT users:

- Servlets increase the speed of the data transfer between the Web browser and D3 because the servlet is loaded only one time by the Web server and is shared among users.
- Servlets are platform independent and easy to use.

The FCServlet interface supplements the existing FCcgi and FCisapi conduits by combining the strengths of CGI's cross-platform support and ISAPI's speed. When invoked by the Web server, FCServlet transfers data from the Web server to FCService. It also receives data from FCService and translates the reply back to the Web server.

FCServlet cannot be used without a third-party Java Servlet engine. The Java Servlet engine is not provided on the FlashCONNECT CD-ROM but is available at no cost from the following URL:

<http://java.sun.com/products/>

For more information on configuring and using the FCServlet, see [Testing FCServlet](#).

JNI Interface

The JNI interface plays a major role in the transaction between the FCServlet and FCService. The JNI interface enables FCServlet, written in Java, and FCService, written in C++, to communicate. There is no direct access to the FlashCONNECT JNI interface. All access is through FCServlet.

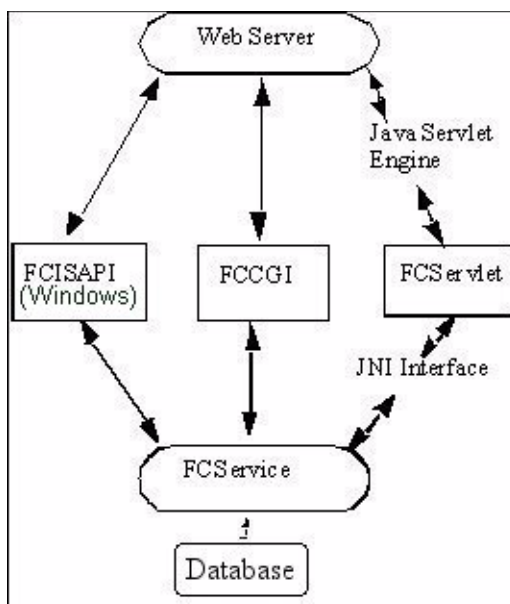
Working with FCServlet

FCServlet provides an alternative path for the flow of data between the Web server and FCService.

The path contains three stages:

1. FCServlet gets the request from the Web browser via the Java Servlet engine and passes the request to the JNI interface.
2. The JNI interface translates the environmental and form variables and passes them to FCService.
3. FCService passes the request to D3 which processes the information and sends back the result along the same path ending at FCServlet. FCServlet then passes the data back to the Web server.

The following diagram illustrates the three possible paths through which the Web server and FCService can communicate.



Testing FCServlet

After installing the FlashCONNECT software, but before using FCServlet, perform tests to confirm that the Java Servlet Engine runs properly and is properly configured to run FCServlet.

There are two tests to perform:

Test 1: Use FCHelloWorld to test the communication between the Web server and the Java Servlet Engine.

Test 2: Use FCJNIHelloWorld to test if the Java Servlet Engine works properly with the Java Native Interface codes.

Test 1

Use test 1 to confirm the Web browser can run a simple Java servlet.

To run this test successfully, the Web server must be running and a Java Servlet Engine must be started.

NOTE— D3 need not be installed and FCService need not be running.

To test the communications between the Web browser and the Servlet engine:

- From the Web browser (Netscape or IE) invoke the following URL:

`http://webservername/servlet/com.rdt.examples.FCHelloWorld`

where *servlet* is a virtual directory that contains available active servlets.

NOTE— It is not necessary that an active servlet be in this virtual directory. Servlets within an archived file (.jar) are acceptable as long as the servlet path is indicated in its property file.

Test Results

The test is successful if:

- A Web page displays with the title "FCHelloWorld."
- The Web page contents, "Hello World!" display.

The test failed if:

- An error message or no message displays.

If the test failed, do the following:

1. Confirm that the Web server supports Java servlets.
2. Confirm that the Java Servlet engine is properly installed.
3. Verify that the correct path was used in the URL.
4. Using other steps in the FCServlet Diagnostic application, confirm the Web server is running. For more information, see [FCServlet Diagnostic](#).

Test 2

Use test 2 to confirm that the Web server can run a simple JNI servlet.

To run this test successfully, the Web server must be running and the Java Servlet Engine must be ready.

NOTE— D3 need not be installed and FCService need not be running.

To test if the Java Servlet Engine works properly with the JNI interface:

- From the Web browser (Netscape or IE) invoke the following URL:

`http://webservername/servlet/com.pick.examples.FCJNIHelloWorld`

where *servlet* is a virtual directory that contains active servlets.

NOTE— It is not necessary that an active servlet be in this virtual directory. Servlets within an archived file (.jar) are acceptable as long as the servlet path is indicated in its property file.

Test Results

The test was successful if:

- A Web page displays with the title "FCJNIHelloWorld."
- The Web page contents, "Hello World From JNI!" display.

The test failed if:

- An error message or no message displays.

If the test failed, do the following:

1. Confirm that the Web server supports JNI servlets.
2. Using other steps in the FCServlet Diagnostic application, confirm that the Web server is running. For more information, see [FCServlet Diagnostic](#).

Using FCServlet

After Test 1 and Test 2 run successfully, run FCServlet to access several D3 applications.

To run FCServlet:

- Issue a URL in the form:

`http://webservername/servlet/
com.rdt.a.bin.FCServlet?w3exec=ApplicationName`

where

<i>webservername</i>	Name of the Web server to run,
<i>servlet</i>	Virtual directory that contains the active servlets.
<i>com.rdt.a.bin.FCServlet</i>	Package name of the FCServlet.
<i>ApplicationName</i>	Name of the FlashCONNECT application.

From here you can tell if FlashCONNECT is running, or if D3 is running, depending on the message you get from running the FCServlet.

FCServlet Diagnostic

The FCServlet Diagnostic tool is a servlet diagnostic application that tests whether a system is capable of running the FlashCONNECT Java Components.

Run the FCServlet Diagnostic to ensure that the FlashCONNECT Java Components are set up properly. In addition, if any of the FlashCONNECT Java Components, such as FCServlet or FlashCONNECT Socket Server Manager, fail to serve the request/response, run the FCServlet Diagnostic to determine the problem.

NOTE— The FCServlet Diagnostic tool is available to Windows and UNIX users that installed the FlashCONNECT Socket Server Manager graphical user interface. Windows users must also have set the FlashCONNECT Socket Server Manager to interact with the desktop.

Using the FCServlet Diagnostic

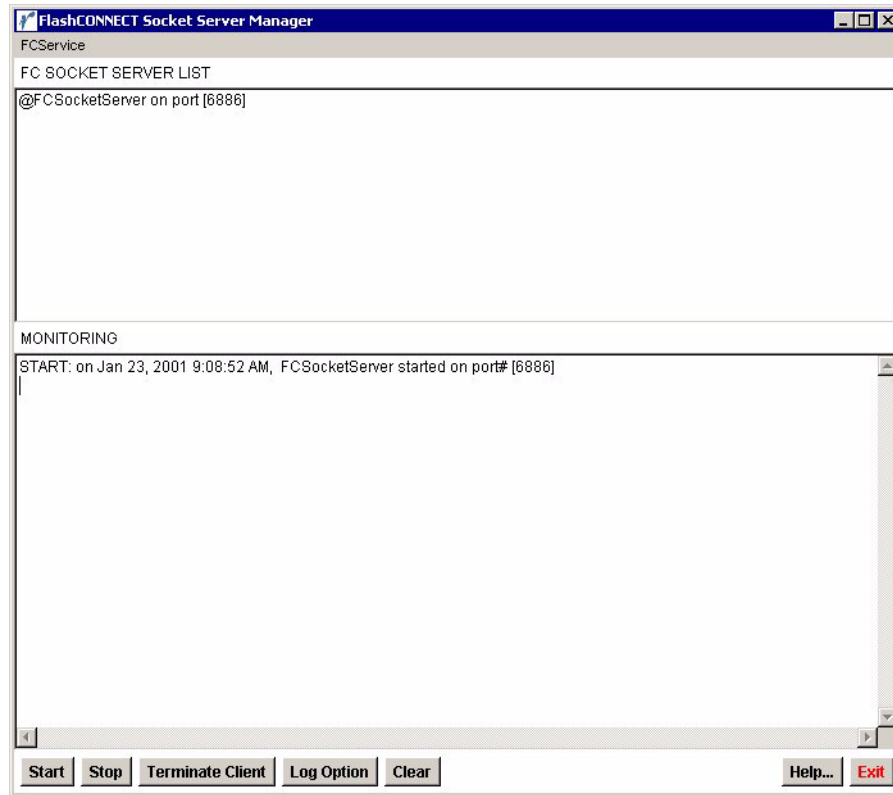
Use the FCServlet Diagnostic dialog box to run tests, view a test report log, reconfigure the setup parameters, reset the test selection, and view information about FCServlet.

The FCServlet Diagnostic tool is nested within the FlashCONNECT Socket Server Manager. To use the FCServlet Diagnostic tool, launch the FlashCONNECT Socket Server Manager first.

To launch FCServlet Diagnostic and set the parameters:

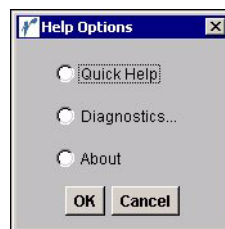
1. Launch the FlashCONNECT Socket Server Manager.

The FlashCONNECT Socket Server Manager dialog box displays.



2. Click Help.

The Help Options dialog box displays.



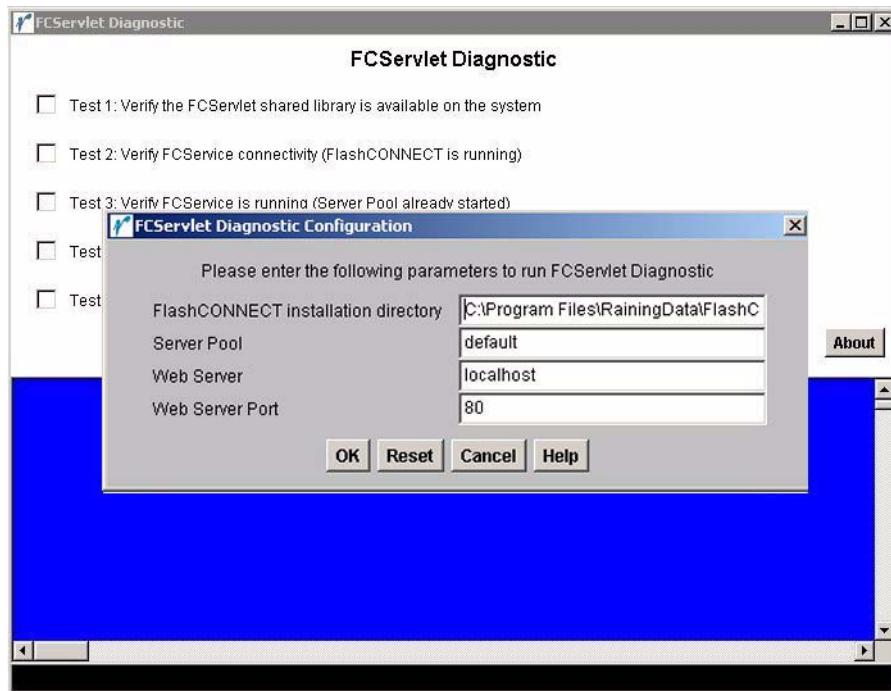
3. Select the Diagnostic option, then click OK.

The Diagnostic Options box displays.



4. Select FCServlet Diagnostic, then click OK.

The FCServlet Diagnostic dialog box and the FCServlet Configuration box display.



The configuration box provides the parameters needed to run the FCServlet Diagnostic.

5. Enter the appropriate parameters, then click OK to return to the FCServlet Diagnostic dialog box.

The FCServlet Diagnostic is ready to test the FlashCONNECT Java components.

Testing

Once the FCServlet Diagnostic is ready, perform any of the five tests to check the compatibility of the system.

NOTE— Tests 1 and 5 can be skipped if only mvDesigner is being used and FCServlet is not being used.

- Test 1 verifies that the FCServlet shared library is available on the system. Run this test to confirm that the FlashCONNECT shared library file, which is a part of FlashCONNECT Java Components, is properly installed.
- Test 2 verifies that the FCService Connectivity exists on the system. Run this test to check if FCService is currently running on the system (confirms whether or not FlashCONNECT is started.)
- Test 3 verifies that FCService is running. Run this test to confirm that FCService is ready and the D3 Server Pool is already started.
- Test 4 verifies that the FC Java Native Interface is working properly. Run this test to ensure that the interface layer between the Java application and FCService is working.
- Test 5 verifies that the servlet add-on engine is available on the system. Run this test to confirm that the third party servlet engine is installed correctly on the system.

The table below lists each test and explains why each test is passed or failed.

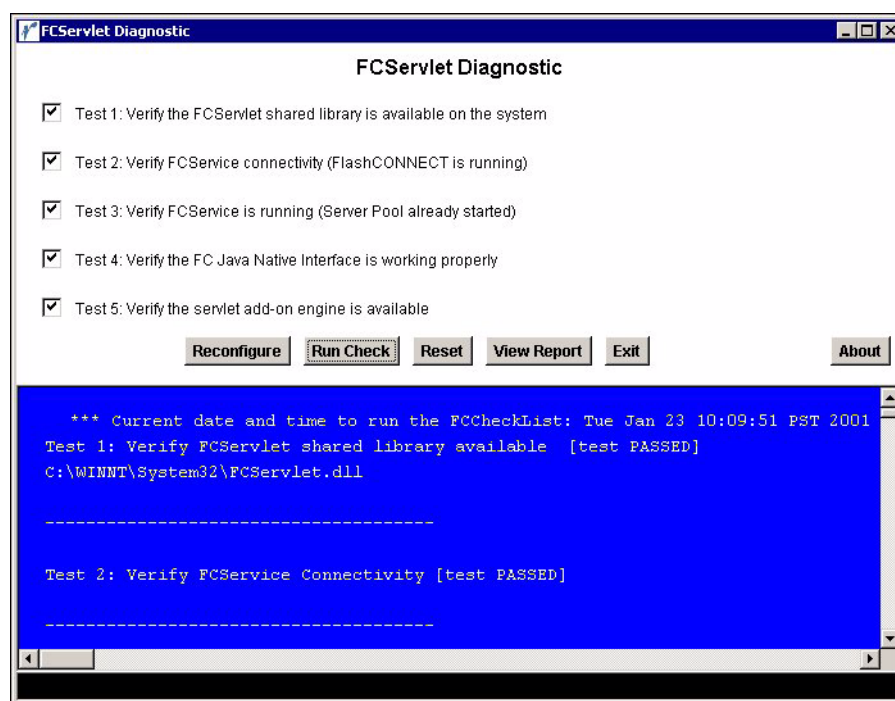
Test Number	Passed	Failed
1	The shared library file is located on the system running FlashCONNECT.	The FCServlet shared library file is not located on the system. The system cannot run the FCServlet.
2	FlashCONNECT is running.	FlashCONNECT is not started yet.
3	D3 is running.	D3 is not started and/or FlashCONNECT is not started yet.
4	The FC Java Native Interface and the FCService can communicate with each other	There is broken communication between the FC Java Native Interface and FCService. Check to see if the shared library is installed. It is usually installed in: For UNIX: /usr/bin/lib/FCServlet.so For Windows: C:\Winnt\system32\FCServlet.dll

Test Number	Passed	Failed
5	A Java Servlet Engine is installed on the system and available.	A Java Servlet Engine is not configured properly to run FlashCONNECT, the Web server is not running, the Web server port number is incorrect, or FlashCONNECT is not running.

To test the FlashCONNECT Java components:

1. Select any of the check boxes to run the tests.

The results display in the FCServlet Diagnostic dialog box.



2. Click Run Check.

After each test completes, a dialog box displays noting whether the test succeeded or failed.

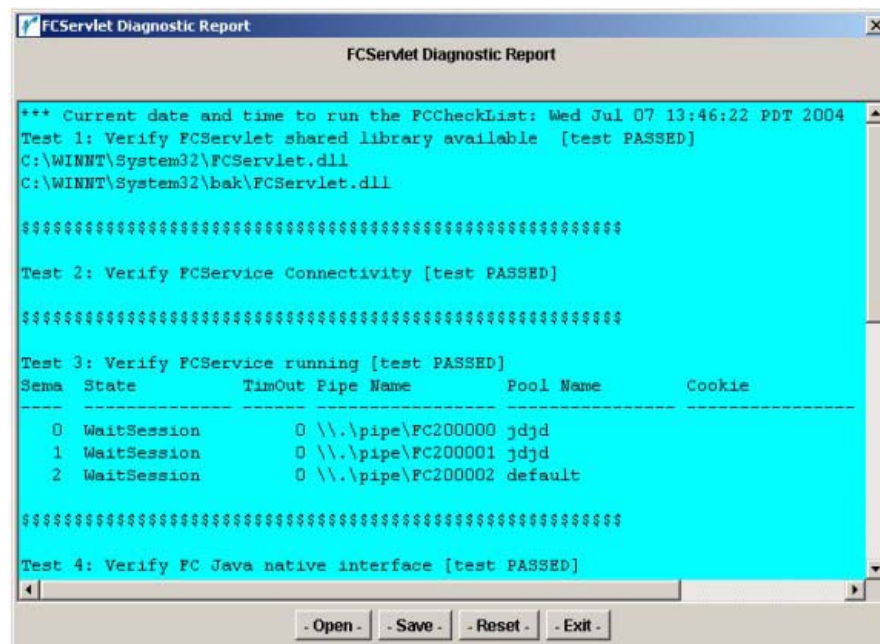
Viewing the FCServlet Diagnostic Report

After running tests, view the test report that describes the test results.

To view the report:

1. Click View Report from the FCServlet Diagnostic dialog box.

The FCServlet Diagnostic Report displays.



2. Do one of the following:

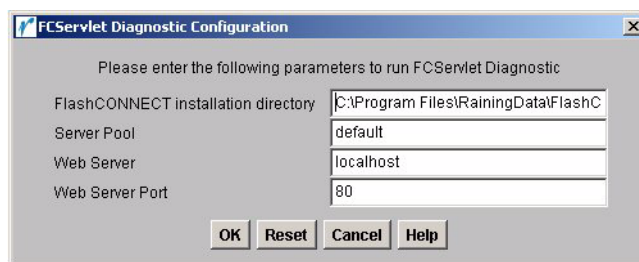
- Click Save to save the report. Enter the file name for the report, then click OK.
- Click Open to view a report previously saved on the system or, click Browse to locate the file, then click OK.
- Click Reset to clear the report without saving the results.
- Click Exit to close the report and return to the FCServlet Diagnostic dialog box without saving the test results.

Reconfiguring FCServlet Diagnostic Parameters

You can reconfigure the parameters to reflect the current setup of FCServlet on the system.

To reconfigure setup parameters:

Click Reconfigure from the FCServlet Diagnostic dialog box. The FCServlet Diagnostic Configuration box displays.



3. Change any of the parameters, then click OK.

Resetting the Test Selection

You can clear the FCServlet Diagnostic dialog box and reselect the tests to perform.

To clear the FCServlet Diagnostic dialog box and select new tests:

1. Click Reset in the FCServlet Diagnostic dialog box.

The test check boxes and the data in the comments section are cleared.

2. Select any of the check boxes next to the tests that you want to perform.
3. Click Run Check.

Viewing Information about the FCServlet Diagnostic Application

You can display the current version and copyright information.

To display version and copyright information:

- In the FCServlet Diagnostic dialog box, click About.

Configuring, Starting and Stopping FlashCONNECT Components

These instructions are for both the UNIX and Windows platforms. When both the Web server and the FlashCONNECT service are properly configured and started, start the FlashCONNECT servers.

Turnkey Installation

A turnkey installation is one in which FlashCONNECT becomes available when the host system starts.

NOTE—A turnkey installation can be created automatically when installing the WWW account by responding yes to the prompt:

Do you want USER-COLDSTART and USER-SHUTDOWN to automatically start and stop FlashCONNECT in the *accountname* account at boot and shutdown time? (Y/N)

See the FlashCONNECT 3.5 and mvDesigner 2.1 Installation Guide for more information.

Complete the steps below to modify the installation to be a turnkey installation.

To create a turnkey installation:

1. Modify the D3 user-coldstart macro to include the steps taken to start the FlashCONNECT components. For example, these lines could be added to the user-coldstart script in the dm account:

```
logto www
clear-jobs (optional)
clear-file dm, runtime-errors, (optional)
clear-file w3lines
WWW-MANAGER START
logto dm
```

Where: *www* is the FlashCONNECT account.

2. Modify the D3 user-shutdown macro to include the steps taken to stop the FlashCONNECT components. For example, these lines could be added to the user-shutdown script (called :SHUTDOWN in mvBase) in the dm account:

```
logto www
WWW-SHUTDOWN
logto dm
```

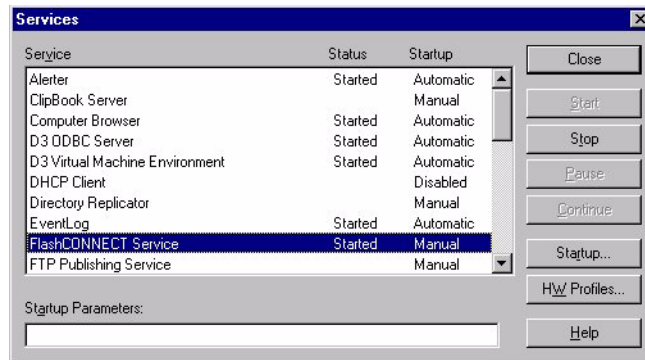
3. Do one of the following:

UNIX: Modify the UNIX /etc/inittab file or the rc.d files to include the steps taken to start the FCService.

Windows: Modify the FlashCONNECT entry in the Service dialog box.

- a. From the Control Panel, select Services.

The Services dialog box displays.



- b. To start FlashCONNECT automatically, double-click FlashCONNECT Service.

The FlashCONNECT Service dialog box displays.



- c. Change the Startup Type to Automatic.
d. Click OK.
e. Click Close in the Services dialog box.

Starting FlashCONNECT in the Foreground

Manually starting FlashCONNECT in the foreground from TCL facilitates the debugging process. Follow these instructions to start FlashCONNECT in the foreground.

To start FlashCONNECT in the foreground:

1. From TCL, enter the command below:

```
WWW-POOL W3SERVERPOOL
```

where *W3SERVERPOOL* is the name of the set of FlashCONNECT server pools.

2. Enter the command below:

```
WWW-IRUN WEB-HOST PORT W3SERVERPOOL
```

where

<i>WEB-HOST</i>	Name or IP address of the Web host running FCSERVICE.
<i>PORT</i>	TCP/IP port number on which FCSERVICE is listening.
<i>W3SERVERPOOL</i>	Name of the Server pool this connection is in.

FlashCONNECT is started in the foreground. To stop FlashCONNECT, log off the port.

NOTE—Run a `list-jobs` from TCL to access more information on the use and syntax of `WWW-IRUN`.

Running FlashCONNECT Ports in the Foreground (D3 Only)

FlashCONNECT can be started on a physical line where there is no physical device (tty). This allows FlashCONNECT ports to be run in the foreground by spawning the ports in the inittab.

To run a port in the foreground in the `/etc/inittab` file, add the line:

```
pick07:23:respawn:d3 -n pick0 7 -u
```

Customizing the Local Environment

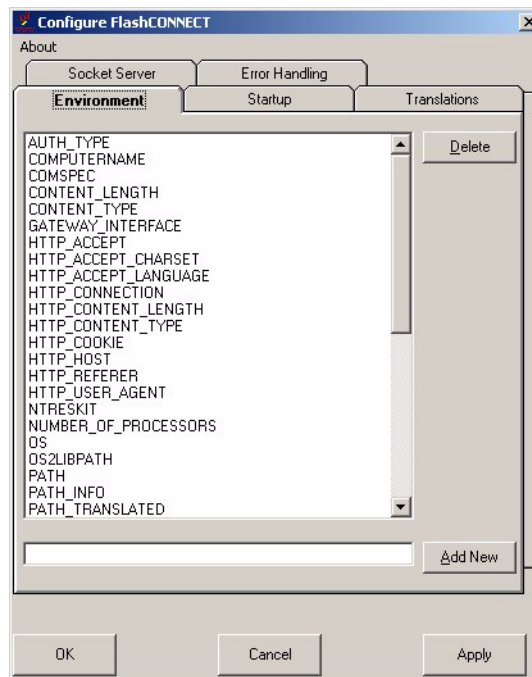
An administration utility is included with FlashCONNECT to customize your local environment. To start the administration utility:

- Select Start > Programs > Raining Data > FlashCONNECT > Configure from the Start menu.

Environment tab

When the FlashCONNECT service passes a request to your application, several environment variables are sent with the request. The Environment tab allows the addition or deletion of environment variables sent to the application. Use this tab to manage overhead.

Environment variables can be added or deleted at any time. The FlashCONNECT service does not need to be restarted for the changes to take effect.



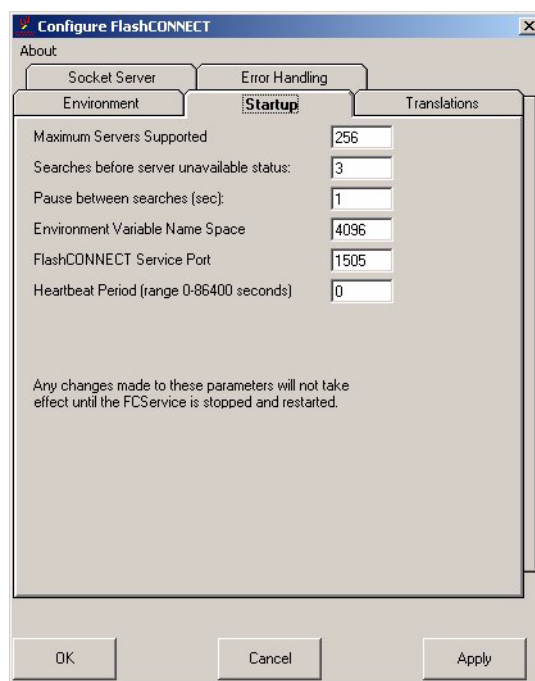
The Environment tab options are:

Option	Description
Delete	Deletes environment variables. Select one or more variable from the list, then click on the Delete button.
Add New	Adds an environment variable. Type the environment variable name in the space provided. Click the Add New button.
OK	Saves changes and exits the dialog box.
Cancel	Quits without saving any unapplied changes.
Apply	Applies changes and stays within the application (when available).

Startup Tab

When the FlashCONNECT service starts, it uses the parameters located in the Startup tab. These parameters are *only* read at startup time. Stop and restart the FlashCONNECT service for changes in these parameters to take effect.

Schedule FlashCONNECT service restarts carefully - all running applications terminate when the FlashCONNECT service is stopped.



The Startup tab options are:

Option	Description
Maximum Servers Supported	<p>Specifies the maximum number of concurrent FlashCONNECT connections.</p> <p>Enter the maximum number of connections to allow. Each increment uses 28 bytes of additional shared memory and may be set as high as desired, subject to the limits of your system. The default is 256 (default is 16 on Linux).</p> <p>NOTE—When all connections are depleted, control functions are no longer accepted by FCSservice, so an additional server slot is automatically created for administrative use.</p>
Searches before server unavailable status	<p>Number of times FCSservice searches for an available connection before ending the search. The default is 3.</p>
Pause between searches (sec)	<p>Time in seconds FCSservice waits between searches for available connections. The default is 1.</p>
Environment Variable Name Space	<p>Sets the upper limit on the number of environment variables passed to D3 from the FlashCONNECT server.</p> <ul style="list-style-type: none"> • Each environment variable uses one byte. • Each letter of the environment variable name uses one byte. • The entire set uses one byte. • The minimum value (and default value) is 4096. <p>For example: If only two environment variables are passed to D3 (HTTP_ACCEPT and AUTH_TYPE) the limit could be set to: 23 (len(HTTP_ACCEPT)+1 + len(AUTH_TYPE)+1 +1)</p>
FlashCONNECT Service Port	<p>TCP/IP port number that the service listens to for D3 FlashCONNECT servers (phantoms) to connect.</p> <p>It can be any valid TCP/IP port number, but must match the port number provided with the <code>WWW-START</code> command.</p> <p>The default is 1505.</p>
Heartbeat Period (range 0-86400 seconds)	<p>Number of seconds FCSservice waits between sending check messages to D3. Range 0 = no heartbeat to 86400. See Configuring the Heartbeat for a discussion about setting this value.</p> <p>The default is 0.</p>

FlashCONNECT Service

The FlashCONNECT service (FCService) was installed during the installation process. On Windows systems, the service can be started from Services located in the Control Panel.

- If the computer was restarted at the end of the installation process, or the FlashCONNECT service was started manually from the Control Panel, the service is ready for operation.
- If the computer was not restarted and the FlashCONNECT service was not started manually, start the FlashCONNECT service from Services located in the Control Panel.

Starting FCService

To start FCService in UNIX using an alternative flashconnect.conf file, enter this command:

```
./fcservice -c path to file
```

UNIX:

To manually start FCService:

1. Go to the directory where FCService is located.
2. Enter:

```
./FCService
```

To automatically start FCService:

Modify the UNIX /etc/inittab file or the rc.d files to include the steps taken to start FCService.

Windows:

To start FCService:

1. Double-click the Services from the Control Panel.

The Services dialog box displays.

2. Select FlashCONNECT Service from the list.
3. Click Startup.

4. Set the Startup Type to Automatic or Manual.

If you set the Startup Type to Automatic, reboot the machine to start. If you set the Startup Type to Manual, click Start.

5. Set the Log On As: parameter.
6. Click OK, then click Close.

Starting FlashCONNECT on D3

To start FlashCONNECT on D3:

1. Log to the www account.

`: logto www`
2. Configure the server pools. See [Connecting Database Server Pools](#) for more information.
3. Start the w3Monitor and all server pools. Enter:

WWW-MANAGER START

Stopping FCService

To stop FCService:

UNIX:

Use the following command to stop FCService:

- `FCDebug -s`

Windows:

1. Double-click the Services icon from the Control Panel.
2. Select FlashCONNECT Service from the list in the Services dialog box.
3. Click Stop, then click Close.

Stopping FlashCONNECT on D3

To stop FlashCONNECT on D3:

1. Log to the www account.
2. Enter:

```
WWW-MANAGER STOP
```

Manually Starting and Stopping the w3Monitor

These instructions are for both the Windows and UNIX platforms. The `WWW-MONITOR` command is executed in the `www` account to manually start and stop the `w3Monitor` as a phantom process. Use the appropriate syntax:

```
WWW-MONITOR START  
WWW-MONITOR STOP
```

NOTE— `WWW-MONITOR stop` works only if the `w3Monitor` was started using `WWW-MONITOR START` or `WWW-MANAGER START`.

Also refer to the topic [Controlling Server Pools and w3Monitor Manually](#). Additional details on `WWW-MONITOR`, including syntax, is located in the [FlashCONNECT Programmer's Online Reference](#).

Customizing the FlashCONNECT Server and Web Server

Instructions for using the FlashCONNECT server and the Web server are discussed. Information provided includes:

- Confirming that the D3 FlashCONNECT connections are established to the Web server.
- Identifying server problems using `WWW-STATUS` and `w3Lines` messages.
- Setting up FlashCONNECT heartbeats if the network connection between the FlashCONNECT Web server and FlashCONNECT D3 server passes through a firewall, or to close a connection if the FlashCONNECT Web server or D3 server terminates.
- Customizing the FlashCONNECT driver application on the database server, and `WWW-IRUN` application use.
- Configuring a Web server (Windows only).

Using WWW-STATUS

Use `WWW-STATUS` to confirm that the D3 FlashCONNECT connections are established to the Web server.

`WWW-STATUS` displays the status of one or more FlashCONNECT lines. Both the FlashCONNECT internal status information and the `where` command information for each line display. If a select list is active, `WWW-STATUS` uses it to allow viewing of selected subsets of the active lines. Examples of `WWW-STATUS` for D3 and `mvBase` are shown below.

In D3:

```

www-status 15:08:29      31 JUL 2002 Page 1

  Idx  Server Pool/      Line/ User/      Time/
      Host Name         Port  Status    Date      Application
-----
    2  default          260
      127.0.0.1        1505 Input      14:34:15
260  019670 FF10 000018    P PX_RESUME:000      07/31/02

    3  default          261
      127.0.0.1        1505 Input      14:22:00
261  01969D FF10 000018    P PX_RESUME:000      07/31/02

    4  default          262
      127.0.0.1        1505 Input      14:20:25
262  0196A3 FF10 000018    P PX_RESUME:000      07/31/02

    5  default          263
      127.0.0.1        1505 Input      14:20:25
263  0196A2 FF10 000018    P PX_RESUME:000      07/31/02

    6  default          264
      127.0.0.1        1505 Input      14:20:25
264  0196A1 FF10 000018    P PX_RESUME:000      07/31/02

The FlashCONNECT Monitor is running on port : 259

Number of FlashCONNECT licensed ports : 7
Number of Ports currently used        : 5
Number of Ports available              : 2
Press <RETURN> to continue...

```

In mvBase:

```
WWW-STATUS 15:03:29      31 JUL 2002 Page 1

  Server Pool/      Line/ User/      Time/
Idx Host Name|      Port  Status      Date      Application
-----
      mvbase              10      15:03:28
0 127.0.0.1      1505 Input      07/31/02

The FlashCONNECT Monitor is running on port : 5

Number of FlashCONNECT licensed ports : 7
Number of Ports currently used       : 1
Number of Ports available            : 6
Press <RETURN> to continue...
```

The information on the first line of each entry is the FlashCONNECT internal status. The information on the second line (or 3rd line in D3) displays the current execution status of processes currently logged on to the system (where command results).

The table below describes each element in the first line of WWW-STATUS.

Element	Description
Idx	ID number to an internal table index on the Web server.
Server Pool	Server pool name.
Host Name	Web server IP address or Host Name.
Line	Socket being used. The first socket is the destination socket on the Web server, the second socket is the source socket on the D3 server.
Port	Port being used.
User	User logged in on the port. Only displays for authenticated applications.

Element	Description
Status	<p>Status of the line.</p> <p>Input: waiting for input.</p> <p>Processing: processing a FlashCONNECT application.</p> <p>Socket: waiting to connect to the Web server.</p> <p>Other: copy the w3Lines item to the screen to see the full text.</p> <p>To see the complete message, enter:</p> <p style="text-align: center;">ct w3lines</p> <p>See Explanation of WWW-STATUS and w3Lines Status Messages for more information.</p>
Time/Date	Time and date of the last time the line was used.
Application	Application that is running as defined in w3Apps.

The second line of each entry (or 3rd line in D3) contains the same information that displays if you were to run the `where` command from TCL. We use the `where` command elements to describe the information presented. Refer to the D3 Reference Manual for additional detail on the `where` command.

The table below describes each element in line 2 of `WWW-STATUS`.

Element	Description
Line	Port issuing the command.
PCB FID	Frame-ID of the primary control block.
PIB Stat	PIB status of the port.
ABS base	Beginning FID of the ABS currently being executed by the port.
Stat	Current status of the port.
R1 & Return stack contents	ABS mode currently executing on the port.

Explanation of WWW-STATUS and w3Lines Status Messages

This section describes in detail the messages shown in the w3Lines file and by the `WWW-STATUS` command.

NOTE— The numeric message value as found in the w3Include item (for example, EQU w3_OK TO 0) and message text as found in the w3Messages and Messages files may change without notice.

The usual sequence of messages for a typical connection is:

1. w3_SettingSocket (Socket)
2. w3_Input
3. w3_ProcessingRequest (Processing)

The last two messages repeat when processing and waiting for requests. Many of these messages display very briefly. Messages displaying for more than a brief period is usually an indication of a problem. Use the information below to identify and resolve any problems.

NOTE— Input is the normal message and is displayed for long time periods.

FlashCONNECT supports messages in several languages. To determine which message name is associated with a specific message text, do the following:

1. Note the message text.
2. Find the matching message text in the system messages or w3Messages file.
3. Note the Item-ID of the item containing the message. The Item-ID has the format: w3xxx or w3xxx111, where xxx is the message number and 111 is an optional language identifier.
4. Look in the wbp file, w3Include item for the message name associated with the message number.
5. Use the table below to determine the meaning and suggested action for each message.

Message Name	Message	Meaning and Suggestions
w3_Input	Input	FlashCONNECT is waiting for a message from FCSservice. The message can be an initial request to run an application or a follow-on page for a persistent connection. This is the most common status for a FlashCONNECT connection.
w3_No Slots	NO SLOTS	FlashCONNECT has contacted FCSservice and FCSservice reported that the maximum number of servers supported by FCSservice has been reached. To resolve this problem, reconfigure FCSservice to support more connections, then stop and restart FCSservice.

Message Name	Message	Meaning and Suggestions
w3_Can't Open Socket	Unable to open Socket (errno=XXX)	An error occurred when using the host operating system. The number shown is the error code returned by the host operating system. See Error Messages and refer to the FlashCONNECT FAQs section of the Raining Data Web site.
w3_BadActivation	Product must be activated before it can be used	FlashCONNECT cannot be activated. Use the <code>sysid</code> command to verify whether FlashCONNECT is activated. <ul style="list-style-type: none"> • If FlashCONNECT is not available, you need a temporary activation. • If you purchased FlashCONNECT, reactivate the D3 System.
w3_BadActivationMask	Activation Mask Invalid	This version of FlashCONNECT is not supported on this platform. Verify that the database server and FlashCONNECT are compatible.
w3_BadClientRead	Read did not complete	The D3 active server received an error while reading a submitted form, or this FlashCONNECT line is being shutdown.
w3_NoUsersAvailable	No D3 licenses available	There were not enough D3 user licenses available to start another FlashCONNECT connection. To correct, do one of these: <ul style="list-style-type: none"> • Free up D3 user licenses being used for other purposes. • Purchase more D3 user licenses. • Configure fewer concurrent FlashCONNECT connections.
w3_ProcessingRequest	Processing	Common status for a FlashCONNECT connection indicating that FlashCONNECT is running an application and it is not waiting for a message from FCService. While in this state, the FlashCONNECT connection does not respond to the <code>WWW-STOP</code> or <code>WWW-MANAGER STOP</code> commands. However, it should respond to a <code>WWW-STOP (F</code> command and it will respond to the <code>WWW-STOP</code> and <code>WWW-MANAGER STOP</code> commands when the request being processed has completed and the connection is ready for the next request.

Message Name	Message	Meaning and Suggestions
w3_SettingSocket	Socket	FlashCONNECT is attempting to establish a connection with FCService. If this message appears for more than a brief period, ensure that the connection to FCService is correct, that FCService is running, and the D3 and Web server components are compatible. This message appears only when a FlashCONNECT connection is starting.

FlashCONNECT Heartbeats

FlashCONNECT heartbeats are often required when the network connection between the FlashCONNECT Web server and FlashCONNECT mvBase server passes through a firewall. As a security feature, some firewalls track a connection's activity and drop the connection if there is no activity for a period of time. Since FlashCONNECT is designed for the efficient use of network resources, when no requests are being made, it does not pass any messages between the FlashCONNECT Web and D3 server. During these idle periods, the firewall may drop the connection between the FlashCONNECT Web and database server. When a request is subsequently made, an error occurs because, unknown to both the FlashCONNECT Web and database server, the connection was dropped. A heartbeat prevents this from occurring by periodically exchanging messages between the FlashCONNECT Web and database server whenever FlashCONNECT is waiting for a message from the Web server.

The heartbeat serves a second purpose as well. If either the FlashCONNECT Web or database server crashes, the server still running does not detect that the other server has stopped, and attempts to continue to use the now non-existent connection. The heartbeat solves this problem by:

- terminating the connection if the database server does not respond to the heartbeat sent by FCService within a few seconds.
- terminating the connection (through w3Monitor) if database is waiting for a request and does not receive a request or heartbeat within a specified time frame.

Whenever a connection is terminated due to a heartbeat failure, a message is logged to the Event log (Windows Web servers), flashconnect.log (UNIX Web servers), or w3Logs,FlashCONNECT file (database Server).

Configuring the Heartbeat

These parameters are set when configuring the FlashCONNECT heartbeat:

frequency	(Set in the server pool/Web host definition from the database server.) Sets the frequency (in seconds) in which w3Monitor checks this configuration.
heartbeat	(Set from the database server.) Sets the server pool/Web host maximum connection time, for the last communication (either a heartbeat or a request) with FCService after which w3Monitor should terminate the connection.
heartbeat period	(Set from the Web server.) Sets the frequency in which FCService sends the heartbeat.

When FCService starts sending a heartbeat, it sends a heartbeat to the first connection, then after the connection responds or is terminated, sends a heartbeat to the next connection. The heartbeat period should be long enough so that it does not cause excessive network traffic and short enough that crashed connections are terminated in a timely period. Additionally, set the heartbeat period and frequency so that connections are not incorrectly terminated. We recommend that:

- the w3Monitor *heartbeat* is at least twice the specified *heartbeat period*.
- the w3Monitor *frequency* is less than the w3Monitor *heartbeat*.

For example, when you configure a server pool from the FlashCONNECT Server Pool Maintenance Web page, set the w3Monitor frequency to 60 and the w3Monitor heartbeat to 1201 as displayed in the following example.


FlashCONNECT Server Pool Maintenance

The FlashCONNECT Server Pool Maintenance screen allows creation, deletion, or modification of Server Pools. A Server Pool defines the connection between the database server running the database component of FlashCONNECT and the server running the external component of FlashCONNECT.

Information about each field is available by clicking on the underlined column headings. Detailed information about each feature is available in the manuals.

Line	<u>Delete</u> Pool	<u>Server Pool</u> *	<u>Host</u> *	<u>Port</u>	<u>Minimum</u>	<u>Maximum</u>	<u>Minimum</u> Free	<u>Maximum</u> Free	<u>w3Monitor</u> Frequency	<u>w3Monitor</u> Heartbeat
1	<input type="checkbox"/>	default	jroth	1505	1	3	0	1	60	1201
	Add									

* = Required


RainingData

File Maintenance
☐ User Groups
☐ Server Pools
☐ Logs
☐ Users
☐ Applications

Reports
☐ Sessions
☐ Concurrent Use
☐ Redirection Use
☐ Rotating Content
☐ Line Status
☐ Contacts

Then, set the Heartbeat Period to 600.

For UNIX: Edit the flashconnect.conf file.

For Windows: Select Start > Programs > Raining Data > FlashCONNECT > Configure from the Start menu.

NOTE— The heartbeat is configured from the FlashCONNECT Server Pool Maintenance Web page. For more information, see [Configuring, Starting and Stopping FlashCONNECT Components](#).

Customizing WWW-IRUN

The FlashCONNECT driver application on the database server consists of the WWW-IRUN and WWW.IRUN2 applications. WWW-IRUN allows any necessary programming to be performed before the FlashCONNECT driver application, WWW.IRUN2, starts. The source to WWW-IRUN is provided as part of the release, however, the source to WWW.IRUN2 is not provided.

FlashCONNECT does not currently use `unnamed common`. If your application uses `unnamed common`, enable `unnamed common` by specifying sufficient `unnamed common` in WWW-IRUN and WWW-CONTINUE to enable applications to use it. Consider:

- Most applications won't use `unnamed common`, resulting in wasted memory.
- The amount of space for variables may be reduced by the amount of space used for unnamed variables.
- WWW-IRUN and WWW-CONTINUE must be compiled using the `mcompile` command. The basic syntax for this command is:

```
mcompile filename Item-ID [Item-ID ...]
```

where *filename* is the file to compile and *Item-ID* is the Item-ID (or list of Item-IDs) to compile.

You can optionally perform a `select` and then run `mcompile filename` from the `select` prompt.

NOTE— `mcompile` is not a TCL command. Thus, `mcompile filename *` does not behave as other TCL commands do (the asterisk is interpreted as an Item-ID, not as a wild-card meaning *all items*).

D3 Only: When `mcompile` completes, run this command to optimize (FlashCompile) these programs:

```
compile filename Item-ID [Item-ID ...] (wo
```

Windows Server Administration

Web Server

Any Web server supported by Windows that supports CGI and/or ISAPI, such as Microsoft Internet Information Server (IIS), Netscape, or Apache, can be used with FlashCONNECT. **The following two steps are *required* and should have been completed as part of the installation procedure. *If they were not performed during installation, complete them now to enable FlashCONNECT to run.***

1. Copy the fccgi.exe and fcisapi.dll files to the cgi-bin or scripts folder for the Web server.
 - If there is no cgi-bin or scripts folder, check your Web server documentation, create the folder and copy the files there.
 - Ensure the folder has execute permissions.
 - Read permissions are not necessary for most servers (and are probably not a good idea).
 - If necessary, use the Web Content Manager to create a virtual directory / cgi-bin to the scripts folder.
2. Point the Web server to the WebContent folder by creating a folder to use the FlashCONNECT [Programmer's Online Reference](#) and examples using one of the alternatives below.
 - Create a virtual folder entry, w3library, pointing to the location FlashCONNECT\WebContent folder.

Consult your Web server documentation for instructions to create a virtual folder /w3library to the WebContent folder.
 - Create a folder under the Document Root folder of the Web server and copy the WebContent files to this folder. Rename the folder w3library (default).

Customizing Internet Information Server (IIS), Netscape and Apache Web Server Error Messages

If you are using an IIS, Netscape or Apache Web server and fccgi.exe or fcisapi.dll, you can customize HTML error messages. For example, common server error messages such as "404 File not found" (indicating the server has not found a matching URL) or "Internal Error 500" (indicating the server encountered an unexpected condition that prevented it from fulfilling the request), can be

customized to fit your Web site and client needs. Refer to your Web server documentation for instructions for changing HTTP error messages.

You can modify or create a new file that contains graphics or better represents your Web site. You may want to provide content that is more descriptive about the possible actions your users can take after receiving the error message.

Redirecting Error Messages

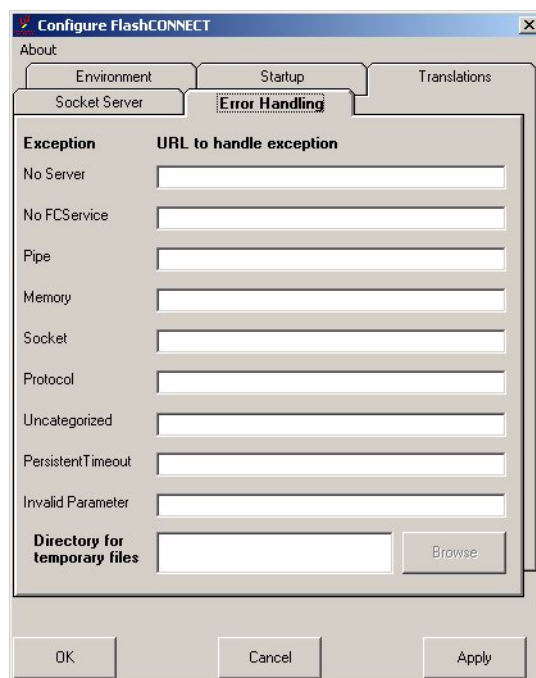
FlashCONNECT allows redirecting categories of error messages to a specified URL, providing the ability to trap error messages on the Web server and apply custom error-handling techniques. The redirection parameters are specified from the FlashCONNECT configuration utility:

For UNIX: The parameters in the flashconnect.conf file are shown below:

```
RedirectParametersPath=  
NoFCServiceException=  
NoServerException=  
PipeException=  
SocketException=  
ProtocolException=  
PersistentTimeoutException=  
InvalidParameterException=  
MemoryException=  
UncategorizedException=
```

For Windows: Select Start > Programs > Raining Data > FlashCONNECT > Configure from the Start menu.

The FlashCONNECT Configuration dialog box displays (parameters are specified from the Error handling tab):



The `RedirectParametersPath` parameter (Directory for temporary files in Windows) specifies the path in which to create a text file that contains the name and value pair parameters of the error as well as the `W3EXCEPTIONERROR` and `W3EXCEPTIONTEXT` variables (generated by the error). The text file is appended and sent to the URL of the specified error.

All other parameters specify the redirect URL for the specified error type.

For example, assume these values are set in the FlashCONNECT configuration file:

```
RedirectParametersPath=/tmp
NoServerException=http://localhost/cgi-bin/nofcserver.pl
```

If `fccgi.exe` and `FCisapi` encounter a `NoServerException` error a text file is created and appended to the URL specified in the `NoServerException` path, resulting in this format:

```
http://localhost/cgi-bin/nofcserver.pl?/tmp/filename.txt
```

where *filename* is a randomly generated 10 character string (for example, jaofvhnwx). The random generation ensures that each file name is unique.

NOTE— • An automatically generated question mark (?) separates the error path and the appended text file.

- In this example, the path specified in the NoServerException actually calls a PERL program. This allows you to create a custom program (PERL or other) for handling the data passed.
- Because each new text file generated is unique, it is good practice to occasionally delete old files from the directory.

Parameter(s)

Parameter (UNIX)	Parameter (Windows)	Description
RedirectParametersPath	Directory for temporary files	Specifies the path in which to create a text file that contains the name and value pair parameters of the error as well as the W3EXCEPTIONERROR and W3EXCEPTIONTEXT variables (generated by the error). This text file is then appended to the URL of the specified error. If left null (default), the text file is not created and nothing is appended to the URL.
NoFCServiceException	NoFCService	Specifies the redirect URL for FCService not running errors.
NoServerException	NoServer	Specifies the redirect URL for server pool not found errors.
PipeException	Pipe	Specifies the redirect URL for pipe errors.
SocketException	Socket	Specifies the redirect URL for socket errors.
ProtocolException	Protocol	Specifies the redirect URL for protocol errors.
PersistentTimeoutException	PersistentTimeout	Specifies the redirect URL for persistent timeout errors.
InvalidParameterException	InvalidParameter	Specifies the redirect URL for invalid ISAPI parameter errors.
MemoryException	Memory	Specifies the redirect URL for memory exception errors.

Parameter (UNIX)	Parameter (Windows)	Description
UncategorizedException=	Uncategorized	Specifies the redirect URL for uncategorized exception errors.

NOTE— If the database detects an error, it calls the w3processerror BASIC program. The source code for this program may be customized and is located in the wbp file.

Example

```
#!/usr/bin/perl
#
# There has been a FlashCONNECT exception redirected here.
# Generate a web page showing all available info.
# Start with the cgi interface
print "content-type: text/html\n\n";

# Now the html tag
print "<HTML>"
print "<HEAD><TITLE>Perl processing of redirection</TITLE></HEAD>\n";
print "<BODY>\n"
print "<P>FlashCONNECT exception redirection!\n";

# Display the env vars if available
$env_file = @ARGV[0];
if ($env_file){
    print "<P>There are some environment variables for clues!\n";
    print "They're in $env_file, and the ones with values are:\n";
    open(FH,$env_file);
    while (<FH>){          # read the next line of the file
        chop;              # remove the line terminator
        $_ =~ / /;         # look for a space in the default variable
        if ($' ne ""){     # if text exists to the right of the space...
            print "<P>\n"; # start a new paragraph
            print $_;      # print the line
        }
    }
    print "\n";
    close(FH);
    unlink $env_file; # delete the file so they do not accumulate
}

# Now the final html tags
print "</BODY>\n";
print "</HTML>\n";
```

Administering Log and JOBS Files

We provide information about administering log files for tracking daily activities and for FlashCONNECT errors. Also included is information for clearing the JOBS file.

Administering Log Files

The FlashCONNECT W3LOGS CONTROL File Maintenance Menu allows creation, deletion, or modification of w3Logs.Control entries. A w3Logs.Control entry controls an associated w3Logs data file. Adding or deleting a w3Logs.Control entry does not create or delete the associated w3Logs data file. Deleting w3Logs data files is done manually. w3Logs data files are created automatically when an entry is written to it.

The FlashCONNECT W3LOGS CONTROL File Maintenance Menu is accessed from the FlashCONNECT File Maintenance Menu. These menus and maintenance screens are also accessible from the www account. For example:

1. Log to the www account from the D3 or mvBase database (if it is not the current account).
2. Type **m** to display the FlashCONNECT Support & Maintenance menu (if the menu is not currently displayed).

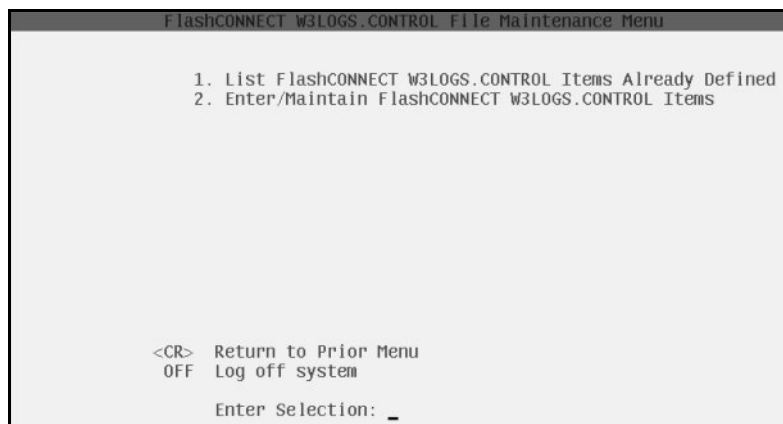
The FlashCONNECT Support & Maintenance menu displays.

3. Select option 4 (Maintain FlashCONNECT Files).

The FlashCONNECT File Maintenance Menu displays.

4. Select option 5 (Maintain FlashCONNECT W3Logs Control Items).

The FlashCONNECT W3LOGS CONTROL File Maintenance Menu displays.



```
FlashCONNECT W3LOGS CONTROL File Maintenance Menu

1. List FlashCONNECT W3LOGS.CONTROL Items Already Defined
2. Enter/Maintain FlashCONNECT W3LOGS.CONTROL Items

<CR> Return to Prior Menu
OFF Log off system

Enter Selection: _
```


Entries are placed in the FlashCONNECT log files by applications using the w3Log subroutine. FlashCONNECT uses several log files. They are:

w3Logs,FlashCONNECT	Tracks daily activity such as starting FlashCONNECT, errors while starting FlashCONNECT, starting and stopping applications, and log on errors.
w3Logs,errors	Used when the w3Log subroutine needs to log an error.
w3Logs,Dump	A diagnostics log file.
w3Logs,w3Rotator	Records the use of w3Rotator.

Logging for each file is individually controlled by an entry in the w3Logs.control file, which has the same name as the corresponding w3Logs data level.

Administering the JOBS File

FlashCONNECT phantoms create entries in the JOBS file. A switch is available that toggles whether the entries remain or are automatically cleared from the JOBS file when the phantom process ends. The default setting is to leave the entries intact.

To set the JOBS file switch:

1. Log to the www account from the D3 or mvBase database (if it is not the current account).
2. Type **m** to display the FlashCONNECT Support & Maintenance menu (if the menu is not currently displayed).

The FlashCONNECT Support & Maintenance menu displays.

```
FlashCONNECT Support & Maintenance

1. Start FlashCONNECT
2. Stop FlashCONNECT
3. FlashCONNECT Status
4. Maintain FlashCONNECT Files
5. Enable FlashCONNECT on Another Account
6. FlashCONNECT Reports
7. Diagnostics
8. Show FlashCONNECT Version Information
9. Activate this version of FlashCONNECT

<CR> Return to Prior Menu
OFF Log off system

Enter Selection:
```

3. Select option 4 (Maintain FlashCONNECT Files).

The FlashCONNECT File Maintenance Menu displays.

```
FlashCONNECT File Maintenance Menu

1. Maintain FlashCONNECT Server Pools
2. Maintain FlashCONNECT Users
3. Maintain FlashCONNECT Applications
4. Maintain FlashCONNECT Groups
5. Maintain FlashCONNECT W3LOGS Control Items
6. Maintain JOBS file (D3 Database only)

<CR> Return to Prior Menu
OFF  Log off system
```

4. Select option 6 (Maintain JOBS file).

NOTE— Option 6 (Maintain JOBS file) is not available from the corresponding FlashCONNECT File Maintenance Menu Web page.

The Maintain JOBS File utility displays.

```
FlashCONNECT phantoms create an entry in the JOBS file.
Select whether the phantom process leaves the entry
in the JOBS file or deletes it from the JOBS file when
the phantom process ends.

Current setting:
A FlashCONNECT phantom does not delete its entry.

Changing the setting affects phantoms that start up
after the change has been made.

1. Toggle setting.

<Enter> Return to Prior Menu

Enter Selection:
```

5. Verify the current setting. For example, the screen above indicates that phantoms do not automatically delete their entries when the process ends.

6. Do one of the following:

- Enter 1 to toggle the current setting.

The Maintain JOBS File utility displays the new setting. Press Enter to exit the utility.

- Press Enter to keep the current setting and exit the utility.

Using Character Translation

Character translations allow the D3 data structure to handle binary or non-ANSI data. The values, 0xFF, 0xFE, 0xFD, and 0xFC are reserved and cannot be used in fields passed into FlashCONNECT. In order to handle these values (non-ANSI data), character translation may be performed on inbound and outbound data.

Performing Character Translation

1 to 1 Translation

Translation Table FF>0F FF<0F FF=0F The following table explains the translation pair and its description.	
Translation Pair	Description
FF>0F	Translates FF to 0F for inbound messages.
FF<0F	Translates 0F to FF for outbound messages.
FF=0F	Translates FF to 0F for inbound messages and 0F to FF for outbound messages. This translation pair is the same as using both FF>0F and FF<0F pairs together.
00>30	<i>Not</i> allowed, cannot translate an incoming binary zero.
22&33	Illegal direction code (&). The line is ignored.

1 to 2 and 2 to 1 Translation

Translation Table +1B FF>0F FF<0F FB=0B The following table explains the translation pair and its description.	
Escape Character	Description
+1B	First line of a 1 to 2 and 2 to 1 translation map file is a '+' followed by an escape character value in hexadecimal. NOTE —1B is used here as an example. You may specify any hex value other than Null (0x00) as the escape value. This implies that the escape character is automatically translated on inbound and outbound messages (1B<->1B1B). It is also implied that any sequence of the escape character followed by an unmapped byte will be translated coming out to just the byte (1Bxx to xx when xx is unmapped.)

FF>0F	Translates FF to 1B0F for inbound messages.
FF<0F	Translates 1B0F to FF for outbound messages.
FF=0F	Translates FF to 1B0F for inbound messages and 1B0F to FF for outbound messages. This translation pair is the same as using both FF>0F and FF<0F pairs together.
00>30	<i>Not</i> allowed. You cannot translate a binary zero.

CAUTION— For example, if the character '0' (0x30) is translated, then all instances of the character '0' to the character '1' (0x31) are translated. This includes the value of w3ServerPool, w3SessionId, w3ProfileId, and characters in HTML tags, such as <SELECT MULTIPLE SIZE=10>, so use caution when selecting escape characters and characters for translation.

Enabling Character Translation

The Character Translation component (located in FCConfig.exe) allows the D3 data structure to handle Unicode data.

UNIX:

The character translation component is part of the installation process. See Installing on a UNIX Platform in the FlashCONNECT Installation Guide for more information.

The FCXlate parameter in the flashconnect.conf file specifies a list of server pool names and paths to translation files used by FCService when processing inbound and outbound data.

```
poolname1,path1;poolname2,path2;...poolnameN,pathN
```

For example:

```
PN1,/etc/pnxlate1.txt;default,/etc/defaultxlate.txt
```

In this example, two server pools are translated; the first server pool is PN1 and it is translated according to the declarations in the file /etc/pnxlate1.txt. The second server pool is the default server pool and it is translated according to the file defaultxlate.txt. Multiple pool names may reference the same translation file.

NOTE— There is no validation performed on the paths. If the specified file does not exist at run time, then no translation module is set up and *no* run-time error is generated.

Each translation file contains a translation map that is used by FCService. This feature operates in two modes:

1 to 1	Translates one byte to one other byte.
1 to 2	Translates one byte to an escape byte followed by another byte.

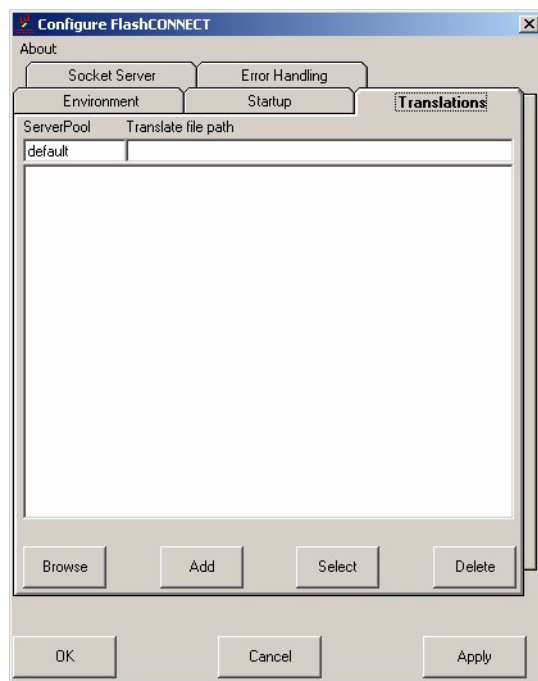
Each line of the map file contains one character translation hex pair separated by a direction code. These pairs must start in column 1 and anything after the translation pair or escape character definition is taken to be a comment.

Windows:

To enable character translation:

1. Select Start > Programs > Raining Data > FlashCONNECT > Configure from the Start Menu.

The Configure FlashCONNECT dialog box displays.



2. Select the Translations tab.
3. Type a name for the server pool in the ServerPool text box.
This field initializes to Default.
4. Type or browse to the translate file path.

For example:

C:\Program Files\FlashCONNECT\FCXlate.txt.

5. Click Add.
6. Repeat steps 3-5 to add additional server pools.
7. (Optional.) Click Apply to save this information to the registry without closing this dialog box.
8. Click OK.

NOTE— Use the Select button to select an existing server pool entry to edit. This also this allows you to add another server pool using the same translation file.

Deleting Translation Server Pools

You can permanently remove a translation server pool.

To remove translation server pools:

1. Select the server pool from the list.
2. Click Delete.

Character Translation Error Messages

If you encounter problems when processing translation definitions, the error messages are logged to the flashconnect.log file if using UNIX or the Event log if using Windows.

The following table describes the character translation error messages:

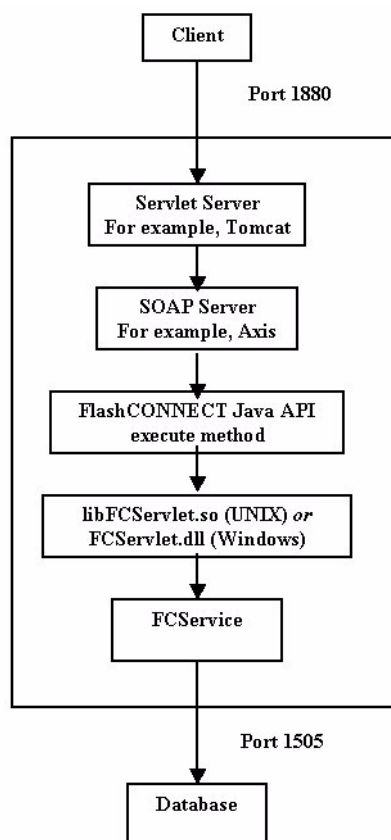
Error Message	Description
Could not open translation map file <i><filename></i> .	Displays when the file containing the translation definition cannot be opened.
Bad or missing translation code <i><invalid character></i> in file <i><translation file name></i> .	Displays if the translation definition contains illegal syntax.
Syntax error in translation specification string <i><line containing the error></i> .	Displays when the translation file syntax is incorrect in the flashconnect.conf file.

FlashCONNECT SOAP Access

This topic discusses deploying FlashCONNECT for SOAP (Simple Object Access Protocol) access.

Deploying the FlashCONNECT SOAP Interface

The FlashCONNECT SOAP interface enables client access to the database via a SOAP server. The procedure involves the client accessing a servlet server (for example, Tomcat). This servlet runs the SOAP server (for example, Axis) that connects to FlashCONNECT. These instructions are SOAP server dependent.



- [Deploying the execute Interface Methods](#)
- [Generating a wsdl File for Client Software \(Optional\)](#)
- [Creating the FlashCONNECT Database Application](#)
- [Accessing the execute Interfaces from Client Programs](#)

Deploying the execute Interface Methods

The execute interface methods (located in the com/rdta/fc/facade/flashconnect.class contained within fcss.jar file) must be deployed. This requires:

- [Preparing for Deployment](#)
- [Creating and Running a Deployment wsdd⁷](#)

Preparing for Deployment

Axis is a servlet that is run by a servlet server (for example, jrun or Apache Tomcat). Therefore, all class paths or shared library paths required by Axis must be set up before starting the servlet server. Any web applications deployed by Axis must be set up as well.

To prepare for deployment:

1. Add the location of the libFCServlet.so file (typically, /usr/lib/rainingdata) to the UNIX LD_LIBRARY_PATH environment variable.
2. Add the fcss.jar file (contains the FlashCONNECT java interface) and any other jar files required by Axis to the CLASSPATH environment variable.
3. Ensure that the fcss.jar file is located in the Tomcat directory structure (required by Apache Tomcat servlet server).

Creating and Running a Deployment wsdd

Create a wsdd for deploying various handlers, chains, and services. There are two methods for accomplishing this:

- `java org.apache.axis.client.AdminClient deploy.wsdd` after the Axis server is running.
- `java org.apache.axis.utils.Admin client|server deploy.wsdd` from the same directory where the Axis engine runs.

The deploy method essentially informs the SOAP server that FlashCONNECT is available for connection.

To create and run the deployment wsdd:

1. Insert the wsdd code. For example:

```
<deployment name="FlashCONNECT" xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
  <service name="flashconnect" provider="java:RPC">
    <parameter name="className"
value="com.rdt.a.fc.facade.flashconnect" />
    <parameter name="allowedMethods" value="execute" />
  </service>
</deployment>
```

2. Run the deployment. For example:

```
java org.apache.axis.client.AdminClient deploy.wsdd
```

NOTE—In case of failures, you may need to create and run a wsdd to undeploy the application. For example:

```
undeploy.wsdd
<undeployment name="FlashCONNECT" xmlns="http://xml.apache.org/axis/wsdd/">
  <service name="flashconnect" />
</undeployment>

java org.apache.axis.client.AdminClient undeploy.wsdd
```

Generating a wsdl File for Client Software (Optional)

These instructions are SOAP server dependent. Below is an Apache Axis wsdl generated for the FlashCONNECT interface. The wsdl language facilitates the client program creation in that it allows you to write less code (than a nonwsdl application requires) to accomplish your program's goals.

NOTE—• The wsdl file contains the name of the server to access.

- `<hostname>` is the name of your server host.
 - 8080 is the default servlet server port.
-

The wsdl file should be either available on your web server to access client software (more flexible) or shipped with the client software (better performance).

```
flashconnect.wsdl
<?xml version="1.0" encoding="UTF-8" ?>
```

```

<wsdl:definitions targetNamespace="http://<hostname>:8080/axis/
  services/flashconnect" xmlns="http://
  schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://
  xml.apache.org/xml-soap" xmlns:impl="http://
  <hostname>:8080/axis/services/flashconnect"
  xmlns:intf="http://<hostname>:8080/axis/services/
  flashconnect" xmlns:soapenc="http://
  schemas.xmlsoap.org/soap/encoding/" xmlns:tns1="http://
  /br.fc.rdta.com" xmlns:tns2="http://
  common.fc.rdta.com" xmlns:wsdl="http://
  schemas.xmlsoap.org/wsdl/" xmlns:wsdlsoap="http://
  schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://
  www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <schema targetNamespace="http://<hostname>:8080/axis/services/
      flashconnect" xmlns="http://www.w3.org/2001/
      XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /
      >
      <complexType name="ArrayOf_xsd_string">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
              wsdl:arrayType="xsd:string[]" />
          </restriction>
        </complexContent>
      </complexType>
    </schema>
    <schema targetNamespace="http://common.fc.rdta.com"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /
      >
      <complexType abstract="true" name="RdtaExceptions">
        <sequence />
      </complexType>
    </schema>
    <schema targetNamespace="http://br.fc.rdta.com" xmlns="http://
      www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" /
      >
      <complexType name="FCClassSetupException">
        <complexContent>
          <extension base="tns2:RdtaExceptions">
            <sequence />
          </extension>
        </complexContent>
      </complexType>
    </schema>
  </wsdl:types>

```

```
<wsdl:message name="executeRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="impl:ArrayOf_xsd_string" />
  <wsdl:part name="in5" type="impl:ArrayOf_xsd_string" />
</wsdl:message>
<wsdl:message name="executeRequest1">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="impl:ArrayOf_xsd_string" />
  <wsdl:part name="in2" type="impl:ArrayOf_xsd_string" />
</wsdl:message>
<wsdl:message name="FCClassSetupException">
  <wsdl:part name="fault" type="tnsl:FCClassSetupException" />
</wsdl:message>
<wsdl:message name="executeResponse">
  <wsdl:part name="executeReturn" type="xsd:string" />
</wsdl:message>
<wsdl:message name="executeResponse1">
  <wsdl:part name="executeReturn" type="xsd:string" />
</wsdl:message>
<wsdl:portType name="flashconnect">
  <wsdl:operation name="execute" parameterOrder="in0 in1 in2 in3
    in4 in5">
    <wsdl:input message="impl:executeRequest" name="executeRequest"
      />
    <wsdl:output message="impl:executeResponse"
      name="executeResponse" />
    <wsdl:fault message="impl:FCClassSetupException"
      name="FCClassSetupException" />
  </wsdl:operation>
  <wsdl:operation name="execute" parameterOrder="in0 in1 in2">
    <wsdl:input message="impl:executeRequest1"
      name="executeRequest1" />
    <wsdl:output message="impl:executeResponse1"
      name="executeResponse1" />
    <wsdl:fault message="impl:FCClassSetupException"
      name="FCClassSetupException" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="flashconnectSoapBinding"
  type="impl:flashconnect">
  <wsdlsoap:binding style="rpc" transport="http://
    schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="execute">
    <wsdlsoap:operation soapAction="" />
    <wsdl:input name="executeRequest">
```

```

<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/
  encoding/" namespace="http://facade.fc.rdt.com"
  use="encoded" />
</wsdl:input>
<wsdl:output name="executeResponse">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/
  encoding/" namespace="http://<hostname>:8080/axis/
  services/flashconnect" use="encoded" />
</wsdl:output>
<wsdl:fault name="FCClassSetupException">
<wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/
  encoding/" namespace="http://<hostname>:8080/axis/
  services/flashconnect" use="encoded" />
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="execute">
<wsdlsoap:operation soapAction="" />
<wsdl:input name="executeRequest1">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/
  encoding/" namespace="http://facade.fc.rdt.com"
  use="encoded" />
</wsdl:input>
<wsdl:output name="executeResponse1">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/
  encoding/" namespace="http://<hostname>:8080/axis/
  services/flashconnect" use="encoded" />
</wsdl:output>
<wsdl:fault name="FCClassSetupException">
<wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/
  encoding/" namespace="http://<hostname>:8080/axis/
  services/flashconnect" use="encoded" />
</wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="flashconnectService">
<wsdl:port binding="impl:flashconnectSoapBinding"
  name="flashconnect">
<wsdlsoap:address location="http://<hostname>:8080/axis/
  services/flashconnect" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Creating the FlashCONNECT Database Application

Below is a sample getquote business rule subroutine and data file.

```
getquote
```

```
001 SUBROUTINE GETQUOTE
002 INCLUDE WBP W3INCLUDE
003
004 *
005 * Sending back a non-numeric for error cases causes
006 * an exception in applications looking explicitly for
007 * numerics. That exception should be handled there.
008 *
009 OPEN "TESTDATA" TO FV_TESTDATA THEN
010 CALL W3GETVAL(SYMBOL,'symbol')
011 READ ITEM FROM FV_TESTDATA,SYMBOL ELSE
012 ITEM = "That symbol does not exist in the TESTDATA file."
013 END
014 END ELSE ITEM = "Can't open TESTDATA file!"
015 CALL W3PRINT(ITEM<1>)
016 CLOSE FV_TESTDATA
017 RETURN

:ct testdata *

IBM
001 92.30

CSCO
001 21.75

XXX
001 99.95
```

Accessing the execute Interfaces from Client Programs

- [Sample Client \(Not Using wsdl\)](#)
- [Sample Client Using wsdl](#)

Sample Client (Not Using wsdl)

1. Compile the sample program. For example, in a command window on a Windows machine using Apache Axis:

```
C:\axis>echo %classpath%
.;c:\axis\lib\axis.jar;c:\axis\lib\commons-logging.jar;c:\axis\lib\COMMONS-
DISCOVERY.JAR;c:\axis\lib\jaxrpc.jar;c:\axis\lib\saa
jar
javac -classpath %CLASSPATH% samples/FCStock/fc.java
```

2. Run the sample program. For example:

```
java -classpath %CLASSPATH% samples/FCStock/fc getquote ibm
```

NOTE— The getquote database business rule is a parameter.

The example below should also execute properly, and display html text:

```
java -classpath %CLASSPATH% samples/FCStock/fc boo ibm
```

The second parameter (ibm) is required by the java program, but is ignored by boo.

```
/*
 * This software may include or be accompanied by software
 * developed by third parties, including the apache software
 * foundation (http://www.apache.org/). Software from third
 * parties is subject to license restrictions. For additional
 * details, please see the Raining Data End User License
 * Agreement and the third party license agreements that
 * accompany the product. Raining Data may modify some third
 * party programs.
 *
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2001-2003 The Apache Software Foundation. All
 * rights reserved.
 *
 * Redistribution and use in source and binary forms, with or
 * without modification, are permitted provided that the
 * following conditions are met:
 *
 * 1.Redistributions of source code must retain the above
 * copyright notice, this list of conditions and the following
 * disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above
 * copyright notice, this list of conditions and the following
 * disclaimer in the documentation and/or other materials
 * provided with the distribution.
 *
 * 3. The end-user documentation included with the
 * redistribution, if any, must include the following
 * acknowledgment: "This product includes software developed by
 * the Apache Software Foundation (http://www.apache.org/)."
 * Alternately, this acknowledgment may appear in the software
 * itself, if and wherever such third-party acknowledgments
 * normally appear.
```

```
*
* 4. The names "Axis" and "Apache Software Foundation" must
* not be used to endorse or promote products derived from this
* software without prior written permission. For written
* permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called
* "Apache", nor may "Apache" appear in their name, without prior
* written permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
* PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR
* ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
* ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*=====
*
* This software consists of voluntary contributions made by
* many individuals on behalf of the Apache Software Foundation.
* For more information on the Apache Software Foundation,
* please see <http://www.apache.org/>.
*/
* This is a copy of axis/samples/stock/GetQuote.java modified
* to use FlashCONNECT.
*/
package samples.FCStock ;

import org.apache.axis.AxisFault;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import org.apache.axis.utils.Options;

import javax.xml.namespace.QName;
import javax.xml.rpc.ParameterMode;
import java.net.URL;

public class fc {
    public String username = "";
    public String password = "";
    public String rulename = "";
```

```
public String serverpool = null;
public String[] parameternames = null;
public String[] parametervalues = null;
// helper function; does all the real work
public String run (String args[]) throws Exception {
    Options opts = new Options( args );
    args = opts.getRemainingArgs();
    if ( args == null ) {
        System.err.println( "Usage: fc getquote <symbol>" );
        System.exit(1);
    }
    rulename = args[0] ;
    parameternames = new String[1];
    parametervalues = new String[1];
    parameternames[0] = "SYMBOL";
    parametervalues[0] = args[1];
    opts.setDefaultURL("http://<your hostname here>:8080/axis/
        servlet/AxisServlet");
    URL url = new URL(opts.getURL());
    /*
    String user = opts.getUser();
    String passwd = opts.getPassword();
    */
    Service service = new Service();
    String res = "";
    Call call = (Call) service.createCall();
    call.setTargetEndpointAddress( url );
    call.setOperationName( new QName("flashconnect", "execute") );
    call.addParameter( "username", XMLType.XSD_STRING,
        ParameterMode.IN );
    call.addParameter( "password", XMLType.XSD_STRING,
        ParameterMode.IN );
    call.addParameter( "rulename", XMLType.XSD_STRING,
        ParameterMode.IN );
    call.addParameter( "serverpool", XMLType.XSD_STRING,
        ParameterMode.IN );
    call.addParameter( "parameternames",
        new javax.xml.namespace.QName("urn:flashconnect",
            "ArrayOf_xsd_string"),
        java.lang.String[].class,
        ParameterMode.IN );
    call.addParameter( "parametervalues",
        new javax.xml.namespace.QName("urn:flashconnect",
            "ArrayOf_xsd_string"),
        java.lang.String[].class,
        ParameterMode.IN );
    call.setReturnType( XMLType.XSD_STRING );
    // call.setUsername( user );
    // call.setPassword( passwd );
```

```
Object ret = call.invoke( new Object[] {
    username,
    password,
    rulename,
    serverpool,
    parameternames,
    parametervalues} );
if ( !(ret instanceof String) ) {
    System.out.println("Received problem response from server:
        "+ret);
        throw new AxisFault("", (String)ret, null, null);
    }
return (String)ret;
}
public static void main(String args[]) {
    try {
        fc fcInstance = new fc();
        String val = fcInstance.run(args);
        // args array gets side-effected
        System.out.println( val);
    }
    catch( Exception e ) {
        e.printStackTrace();
    }
}
};
```

Sample Client Using wsdl

This example program performs the same functions as the previous program, but several of the setups prior to the `call.invoke()` statement are no longer needed (and have been commented out) as a result of using wsdl.

```
/*
 * This software may include or be accompanied by software
 * developed by third parties, including the apache software
 * foundation (http://www.apache.org/). Software from third
 * parties is subject to license restrictions. For additional
 * details, please see the Raining Data End User License
 * Agreement and the third party license agreements that
 * accompany the product. Raining Data may modify some third
 * party programs.
 *
 * The Apache Software License, Version 1.1
 *
 * Copyright (c) 2001-2003 The Apache Software Foundation. All
```

```
* rights reserved.
*
* Redistribution and use in source and binary forms, with or
* without modification, are permitted provided that the
* following conditions are met:
*
* 1.Redistributions of source code must retain the above
* copyright notice, this list of conditions and the following
* disclaimer.
*
* 2. Redistributions in binary form must reproduce the above
* copyright notice, this list of conditions and the following
* disclaimer in the documentation and/or other materials
* provided with the distribution.
*
* 3. The end-user documentation included with the
* redistribution, if any, must include the following
* acknowledgment: "This product includes software developed by
* the Apache Software Foundation (http://www.apache.org/)."
* Alternately, this acknowledgment may appear in the software
* itself, if and wherever such third-party acknowledgments
* normally appear.
*
* 4. The names "Axis" and "Apache Software Foundation" must
* not be used to endorse or promote products derived from this
* software without prior written permission. For written
* permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called
* "Apache", nor may "Apache" appear in their name, without prior
* written permission of the Apache Software Foundation.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
* PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
* APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR
* ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
* ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*=====
*
* This software consists of voluntary contributions made by
* many individuals on behalf of the Apache Software Foundation.
```

```
* For more information on the Apache Software Foundation,
* please see <http://www.apache.org/>.
*/
* This is a copy of axis/samples/stock/GetQuote.java modified
* to use FlashCONNECT.
*/
package samples.FCStock ;

import org.apache.axis.AxisFault;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import org.apache.axis.utils.Options;

import javax.xml.namespace.QName;
import javax.xml.rpc.ParameterMode;
import java.net.URL;

public class fc2 {
    public String username = "";
    public String password = "";
    public String rulename = "";
    public String serverpool = null; // defaults to "default"
    public String[] parameterNames = null;
    public String[] parameterValues = null;
    // helper function; does all the real work
    public String run (String args[]) throws Exception {
        Options opts = new Options( args );
        args = opts.getRemainingArgs();
        if ( args == null ) {
            System.err.println( "Usage: fc2 getquote <symbol>" );
            System.exit(1);
        }
        rulename = args[0] ;
        parameterNames = new String[1];
        parameterValues = new String[1];
        parameterNames[0] = "SYMBOL";
        parameterValues[0] = args[1];
        String res = "";
        /* Define the service QName and port QName */
        /******
        QName servQName = new QName("flashconnect","flashconnect");
        QName portQName = new QName("flashconnect","execute");
        /* Now use those QNames as pointers into the WSDL doc */
        /******
        Service service = new Service( new
            URL("file:flashconnect.wsdl"), servQName );
        Call call = (Call) service.createCall( portQName, "execute" );
        */
    }
}
```

```

String user = opts.getUser();
String passwd = opts.getPassword();
*/
/* done by wsdl above
opts.setDefaultURL("http://<your hostname here>:8080/axis/
    servlet/AxisServlet");
URL url = new URL(opts.getURL());
call.setTargetEndpointAddress( url );
call.setOperationName( new QName("flashconnect", "run") );
call.addParameter( "username", XMLType.XSD_STRING,
    ParameterMode.IN );
call.addParameter( "password", XMLType.XSD_STRING,
    ParameterMode.IN );
call.addParameter( "rulename", XMLType.XSD_STRING,
    ParameterMode.IN );
call.addParameter( "serverpool", XMLType.XSD_STRING,
    ParameterMode.IN );
call.addParameter( "parameternames",
    new javax.xml.namespace.QName("urn:flashconnect",
        "ArrayOf_xsd_string"),
    java.lang.String[].class,
    ParameterMode.IN );
call.addParameter( "parametervalues",
    new javax.xml.namespace.QName("urn:flashconnect",
        "ArrayOf_xsd_string"),
    java.lang.String[].class,
    ParameterMode.IN );
call.setReturnType( XMLType.XSD_STRING );
*/
// call.setUsername( user );
// call.setPassword( passwd );

Object ret = call.invoke( new Object[] {
    username,
    password,
    rulename,
    serverpool,
    parameternames,
    parametervalues} );
if ( !(ret instanceof String) ) {
    System.out.println("Received problem response from server:
        "+ret); throw new AxisFault("", (String)ret, null,
        null);
}
return (String)ret;
}
public static void main(String args[]) {
    try {
        fc fcInstance = new fc();

```

```
String val = fcInstance.run(args);  
// args array gets side-effected  
System.out.println( val);  
}  
catch( Exception e ) {  
e.printStackTrace();  
}  
}  
};
```

Debugging and Diagnosing FlashCONNECT Configuration

Strategies, diagnostics and instructions to help troubleshoot your Windows or UNIX installation are included in these topics:

- [Diagnosing FlashCONNECT Configuration Problems \(D3 Only\)](#)
- [Running the Loop-Back Test \(D3 Only\)](#)
- [Roving Tandem Debugger](#)
- [Application Debug Strategies](#)
- [Determining if the Application is in the Debugger](#)
- [Using the FlashCONNECT Server Debug Program](#)
- [Debugging Server Pools](#)
- [Debugging w3Monitor](#)
- [Recording Outgoing HTML Code and Incoming Variables](#)
- [Logging Diagnostic Messages for FCService and FCSocketServer](#)
- [Examining CGI and Form Variables](#)
- [Diagnostic Messages Displayed On the Browser](#)
- [Determining Release Levels](#)

For topics not covered here, refer to the [FlashCONNECT FAQs](#) section of the Raining Data Web site.

NOTE— Maintain the dm.jobs, file regularly as FlashCONNECT creates an entry in the jobs file when additional connections are created. An undersized dm.jobs, file can cause system degradation.

Diagnosing FlashCONNECT Configuration Problems (D3 Only)

The checkup program examines the overall FlashCONNECT configuration and checks for potential and fatal problems. The checkup program does not change the D3 or FlashCONNECT configuration.

There are two ways to run the diagnostic:

- From TCL type:
run bp checkup
- From the TCL FlashCONNECT Main Menu, select 7. Diagnostics, then select 1. Run the Online Diagnostics.

When the checkup program starts, the following screen displays:

```
This program performs simple checks to uncover problems which
commonly cause FlashCONNECT to not work. This program displays as a
series of questions and answers. Other than starting this program the
user does not enter any information.

Problems which will probably cause FlashCONNECT to not work or not
work with a specific host are labeled 'FATAL'.

Problems which might cause problems while FlashCONNECT is running are
marked, 'NO'.

Tests which pass are marked with 'YES'.

These tests run in a few seconds, do not change the FlashCONNECT
configuration and can be used on a running FlashCONNECT system.

See the FlashCONNECT documentation for more information about each
test.

Press 'Q' to quit or 'Enter' to continue?
```

If Q is entered, the program stops without performing any checks.

The report displays fatal errors which indicate that the tested feature will not work:

- Yes indicates the tested feature should work.
- No indicates the tested feature should partially work.

```
Starting Diagnostic
Checking if FlashCONNECT is activated?           Yes
Are there user licenses available?               Yes ... 50 available
Are there phantom processes available?           Yes ... 26 available
Verifying that the database version is supported Yes ... <version>

Can FlashCONNECT connect to the configured hosts?
FC Host      Port
Local host   1505          FATAL...Could not
                                connect.
172.16.250.218 1505       Yes
eng.rainingdata.com 1505   Yes

Are there enough available phantoms to run the:
Minimum number of connections           Yes ... Requested 10
Maximum number of connections           No ... Requested 60

Are there enough available licenses to run the:
Minimum number of connections           Yes ... Requested 20
Maximum number of connections           No ... Requested 60

Do the host names appear valid?
172.16.250.218                          Yes
host.rainingdata.com                     Yes
Local host                              No
Done
```

Message	Example Values	Comments
Starting		The program is starting the evaluation.
Checking if FlashCONNECT is activated.	Yes FATAL ... not activated	FlashCONNECT must be activated before it can be used.
Are there user licenses available?	Yes ... 5008 available FATAL no free licenses	Calculates the number of currently available D3 user licenses. If there are no available D3 user licenses FlashCONNECT cannot run. Either take action to free up user licenses or purchase more licenses.
Are there phantom processes available?	Yes ... 21 available FATAL no free phantoms	Calculates the number of currently available phantom processes. If there are no available phantom processes D3 cannot run. Either reconfigure D3 making more phantom processes available or take other action to free up phantom processes.
Can this version of D3 run FC?	Yes ... <version> FATAL unsupported version ... <version>	FlashCONNECT requires a minimum D3 release level to run. To run FlashCONNECT, D3 must be upgraded.
Can FlashCONNECT connect to the configured hosts? FC Host Port	<ul style="list-style-type: none"> • FC Host Port • 172.16.250.218 1505 Yes • host.rainingdata.com 1505 Yes • local host 1234 FATAL ... could not connect. • FATAL ... could not create socket. • No ... Cannot open w3Config file • No configuration found 	<p>Each host/server pool combination defined in the w3Config file is checked to verify that the indicated connection can be reached. This indicates that the host and port can be reached and there is a process servicing the port. It does not verify the listening process is FCService. In addition, the ability to create a socket is verified.</p> <p>If no configuration is found or the w3Config file cannot be opened, and there are no fatal errors, FlashCONNECT can be manually started and stopped.</p> <p>Server pool names are checked for validity. An error message displays the invalid server pool name as well as the formatting rules for server pool names.</p>

Message	Example Values	Comments
Are there enough available phantoms to run the minimum number of connections?	Yes ... requested <count> No ... requested <count>	Number of minimum connections for each server pool/host/name combination is totaled and compared to the number of currently available phantom processes. If there are sufficient available phantom processes to run the minimum number of connections, <i>Yes</i> is displayed. If there is an insufficient number of available phantom processes to run the minimum number of connections, <i>No</i> is displayed. The number of minimum requested connections is displayed.
Are there enough available phantoms to run the maximum number of connections?	Yes ... requested <count> No ... requested <count>	Number of maximum connections for each server pool/host/name combination is totaled and compared to the number of currently available phantom processes. If there are sufficient available phantom processes to run the maximum number of connections, <i>Yes</i> is displayed. If there is an insufficient number of available phantom processes to run the maximum number of connections, <i>No</i> is displayed. The number of maximum requested connections is displayed.
Are there enough available licenses to run the minimum number of connections?	Yes ... requested <count> No ... requested <count>	Number of minimum connections for each server pool/host/name combination is totaled and compared to the number of currently available D3 user licenses. If there are sufficient available D3 user licenses to run the minimum number of connections, <i>Yes</i> is displayed. If there is an insufficient number of available D3 user licenses to run the minimum number of connections, <i>No</i> is displayed. The number of minimum requested connections is displayed.

Message	Example Values	Comments
Are there enough available licenses to run the maximum number of connections?	Yes ... requested <i><count></i> No ... requested <i><count></i>	Number of maximum connections for each server pool/host/name combination is totaled and compared to the number of currently available D3 user licenses. If there are sufficient available D3 user licenses to run the maximum number of connections, <i>Yes</i> is displayed. If there is an insufficient number of available D3 user licenses to run the maximum number of connections, <i>No</i> is displayed. The number of maximum requested connections is displayed.
Do the host names appear valid?	172.16.250.218 Yes host.rainingdata.com Yes Local host No	Each host name is examined. If the host name contains a space, it is considered invalid. Otherwise, the host name is considered valid.
Done		The program has finished the evaluation.

Running the Loop-Back Test (D3 Only)

This diagnostic test periodically runs the FlashCONNECT boo application and records in the w3logs,loopback file the state of the application (success or failure), when the test started, and whenever the test switches between success and failure. In addition, the results of each test are displayed on the console. This test can be used to determine when a site goes down or to periodically poll a site.

To run this test you need to know:

- the name of the Web server.
- the port number the Web server is using (usually port 80).
- the number of seconds between checks.

This diagnostic test can be run on a running FlashCONNECT site and does not change any FlashCONNECT configuration parameters. This program can also be copied to and run on D3 systems that do not have FlashCONNECT installed on them.

After starting, loop-back continues running until a key is pressed. Press **q** to quit or Enter to continue.

Respond to these prompts:

Enter the host name: ?	Type the name of the Web server running FlashCONNECT.
Enter the port to use (usually 80): ?	Assign a port number from 1-65535.
Enter number of seconds between checks: ?	Enter any number of seconds from 1-86400.

Roving Tandem Debugger

A characteristic of persistent applications is that they can run on any phantom port running FlashCONNECT. This is not problematic for small development systems that use only one active FlashCONNECT connection. However, on larger multiuser systems, determining the port on which the application being debugged is running can be a repetitive, time-consuming task.

Roving tandems allow a developer to register as the client of an application, and when the application dictates, allows the developer to automatically tandem to the application from the ports that the application is running.

Roving tandem uses:

- `WWW-TANDEM`, a command that registers the developer as the client of an application and then waits for the application to request a connection.
- `w3Tandem`, an API call that registers the application and connects to any waiting developer.

When the developer registers, a record is written to the `w3TandemControl` file with the application name and the port number on which the developer is waiting. When the application registers, it adds its port number to the control record. When both port numbers are known, the `WWW-TANDEM` command connects the developer to the waiting application.

The developer and the application can end the connection. If the application is finished, perhaps having sent information to the user and now stopping until the next page is submitted, it uses the `w3Tandem` API call to temporarily disconnect the tandem to the developer's port. The `WWW-TANDEM` command waits for the application to return, or for the developer to stop the `WWW-TANDEM` session. When the application runs again, possibly from another port, the tandem between the two ports is automatically re-established.

The following code example provides a simple tandem test:

```
bp 'w3mvDTandemTest' size = 24
01 subroutine w3TandemTest
02 include wbp w3include
03
04 call w3Tandem("test", "Start")
05
06 call w3HtmlInit("", "")
07 call w3HtmlHeader("", "")
08 for i = 1 to 10
09     CRT i
10     call w3Printn("<BR>":i)
11     sleep(1)
12 next i
```

```
13 call w3HtmlFooter("", "")
14
15 call w3Tandem("test", "Stop")
16 return
```

If you are using the command line, type this command at the prompt:

WWW-TANDEM TEST

For more information about the w3Tandem API call and the WWW-TANDEM command, see the FlashCONNECT [Programmer's Online Reference](#).

Application Debug Strategies

Static Applications

Static FlashCONNECT applications are normal database applications. Employ standard debugging procedures.

Dynamic Applications

The following methods can be used to debug dynamic applications.

Using the w3Logs File

Within the application, use the w3Log subroutine to log important information to the w3Logs file.

Step-by-Step Debugging

1. Within the application, display diagnostic information to the screen.
 - Use `tandem` or `WWW-TANDEM` to connect to the phantom process running the application and observe what is happening.
 - Leave the diagnostic messages in production code to assist debugging at client sites.
 - When the w3Debugging tool is enabled, the incoming values of the request are displayed but the CGI variables are *not*. When a reply is sent, the outgoing data is displayed. See [Enabling and Disabling the w3Debugging Tool](#).
 - The run-time impact is limited to the time it takes to display the diagnostic messages to the non-existent terminal.
2. Within the application, use the Pick BASIC debug command to place the application into the debugger.
 - Use `tandem` or `WWW-TANDEM` to connect to the phantom.
 - Use standard debugging techniques to debug the application.
3. Start `WWW-IRUN` from TCL and debug the application as you would any other application. Since the options for `WWW-IRUN` change with each release, the options can be determined using the `list-jobs` command to view the phantom processes started by `WWW-START`.

Determining if the Application is in the Debugger

For D3: Use the `WWW-STATUS` command to determine if the application is in the debugger. There are two indicators that you are in the debugger:

- If the stat column displays a D.
- If the second line displays a mode name starting with any of the following:
`PX_DB`, `PX.DB`, `DB` or `DB_GETBUF`.

Using the FlashCONNECT Server Debug Program

The FlashCONNECT server debug program, `FCdebug`, is useful for resolving problems with the FlashCONNECT server. Running the `FCdebug` program from the Web server command line dumps the information in the shared memory segment into a tabular format and displays:

- Server table entries for established connections in two different formats.
- Environmental parameters passed into the FlashCONNECT application.
- Header information.

Locked table entries can be unlocked using `FCdebug`.

The `FCdebug` options are:

Option	Description	
-?	Displays version and options for the <code>FCdebug</code> command.	
-h	Displays FlashCONNECT header information:	
	Column	Definition
	FCProtocol Version	FlashCONNECT internal protocol being used.
	D3ServerSlots*	Maximum number of connections supported.
	ENVSize	Maximum amount of memory available to maintain FlashCONNECT connections.
	FCPort	TCP/IP port for requests to start new FlashCONNECT servers.
	D3Servers_offset	Internal use.
	FCHeartbeat	Number of seconds between the Web server sending the D3 process a Heartbeat.
	env_vars_sema	Internal use.

Option	Description	
	FCIsUp	FlashCONNECT is running.
	FCServicePID	Internal use.
	Semaphores per ID	Internal use. (UNIX only)
	* Configured using the FlashCONNECT Administration program on Windows Web servers or editing FCService.conf on UNIX Web servers.	
-k <code>pool</code>	where <code>pool</code> is the server pool name. Terminates all connections in the specified server pool. However, if the database side is still running, it will reconnect. For example, <code>FCDebug -kdefault</code> closes all connections between FCService and a specific server pool.	
-s	Stops the FCService utility.	
-t	Displays FlashCONNECT Windows server table entries. There is one table entry for every established FlashCONNECT connection.	
	Table Entry	Definition
	Sema	Semaphore number. Also the index in <code>w3SessionId</code> .
	State	<p>Starting: Transitional state.</p> <p>WantW3HostName: Transitional state.</p> <p>WaitSession: Most common state.</p> <p>WaitConnection: Session started, waiting for reply.</p> <p>WaitDB: Waiting for database application. If in this state for an extended period of time, the application is probably in the debugger or has aborted.</p> <p>StartConnection: Used while establishing a connection.</p> <p>StartSession: Used while establishing a session.</p> <p>Timed-Out: Server timed out a connection and is in the process of shutting it down.</p> <p>Stopping: Coming down.</p> <p>SendPulse: Need to send the pulse to the API.</p> <p>WaitDBPulse: Pulse has been sent to the API; waiting for a return pulse.</p> <p>Please note that transitional states are rarely seen.</p>

Option	Description		
	TimeOut	Number of seconds until the connection times out.	
	Pipe Name	Name of pipe used by the ISAPI or CGI connector.	
	Host Name	Value of the w3ServerPool used to establish this connection.	
	Cookie	Value of the w3SessionId for this session.	
	xidx	Index for the translation module. Multiple translation modules may be specified in the configuration file. They are stored starting with an index of 0. An xidx value of -1 indicates that no translation is taking place for that connection.	
	xRsz	Number of bytes required to add to the returned Web page in order to have the correct TCP/IP header count. This is only used during translation and only if 2-to-1 translation is taking place when a value is returned from FlashCONNECT.	
-tr	Displays the FlashCONNECT raw table information.		
	Comma delimited field	Name	Meaning
	1	State	0 = Entry unused. Entries with this state are not shown in the -t view. 1 = Starting 2 = Want W3Host name 3 = Wait session 4 = Wait connection 5 = Wait DB 6 = Start connection 7 = Start session 8 = Timed-out detailed Information about the state field is provided in the -t option. 9 = Stopping 10 = Send pulse 11 = WaitDBPulse

Option	Description		
	2	Sema	Semaphore number. Also the index in w3SessionId.
	3	TimeOut	Number of seconds until the connection times out.
	4	KillInfo	Windows: socket UNIX: Process ID
	5	Pipe Name	Name of the pipe used by the ISAPI or CGI connector.
	6	Host Name	Value of the w3ServerPool used to establish this connection.
	7	Cookie	Value of the w3SessionId for this session.
-uNN	Unlocks table entry NN. For development use only. Before using this option, shut down and restart the FlashCONNECT service.		
-v	Displays the list of environmental parameters passed into the FlashCONNECT application. <ul style="list-style-type: none"> • These variables are retrievable using the w3GetCGI program. • Using the FlashCONNECT Administration program configures the list of passed-in variables on Windows; edit FlashCONNECT.conf on UNIX Web servers. 		
-V	Displays the current version of the FCDebug program.		
-pNN	Sets the protocol version to NN. Not for general use. The FlashCONNECT service must be restarted after using this option.		

Debugging Server Pools

Forcing a connection to a specific port is a useful debugging method for database server pool problems. This method allows you to know in advance which database line the FlashCONNECT session will use.

1. Create a temporary server pool containing only one FlashCONNECT connection.

There should be only one database line supporting this server pool.

2. Use the w3ServerPool parameter to direct the application to that server pool when the application is run.

3. Use the `tandem` command to connect to and debug the FlashCONNECT application.
4. Tandem to the port before running the application, so when the connection is made, the application is already being observed.

Debugging w3Monitor

When w3Monitor detects a FlashCONNECT port in either the system or Pick BASIC debugger, it removes the connection automatically unless this feature is overridden by enabling debugging.

Port x in debugger, connection removed

For more information, refer to the options listed in the topic [w3Apps](#) in the FlashCONNECT Programmer's Guide; or in *w3DebugOK API* found in the FlashCONNECT [Programmer's Online Reference](#) to modify this feature.

CAUTION—If you are debugging a program in FlashCONNECT and heartbeats are being used, the heartbeat may not be processed. If w3Monitor is running and the heartbeat is not processed, the w3Monitor may log off the program being debugged. To avoid this, turn on FlashCONNECT debugging as described in [Using the FlashCONNECT Server Debug Program](#).

NOTE— If D3 performance slows while w3Monitor is checking server pool connections, verify that the `dm.jobs`, and run-time errors files are properly sized. If it is not properly sized, correct the size or delete the completed jobs (use the `clear-jobs` command).

Recording Outgoing HTML Code and Incoming Variables

FlashCONNECT supports the ability to record the HTML information sent to the browser and all of the variables returned from a Web page. This ability facilitates resolving intermittent and complicated problems. The w3Debugging tool controls FlashCONNECT's capability to record in the `w3Logs,Dump` file, outgoing HTML pages and incoming environment and Web form variables.

Enabling and Disabling the w3Debugging Tool

There are three methods that can be used to enable or disable the w3Debugging tool. Although each method has a slightly different scope, they can be used together.

1. Within the URL:

<http://yourwebhost.rainingdata.com/cgi-bin/fccgi.exe?w3exec=boo&w3Debug=1>

Debugging starts when this URL is received. Debugging stops when reset or when the application ends by returning to WWW-IRUN.

2. Call w3DebugOK

w3DebugOK controls caching of templates and substitution arrays, turns on and off incoming data and outgoing forms dumping, and controls restarting a port in the debugger with w3Monitor. See the *D3 Reference Manual* for additional information about this routine.

3. Within the w3Apps file, add the letter *d* to the options value. For example:

Option: d

Debugging starts when the application is started. Debugging stops when reset or when the application ends by returning to WWW-IRUN.

Entries are written to w3Logs,Dump in pairs. The first entry shows the outgoing HTML code and the second entry shows the incoming form/CGI variables. Under some circumstances, either entry can be empty. This also displays incoming and outgoing information to the D3 screen.

In addition, the incoming form values of the request are displayed but the variables are *not* displayed. When a reply is sent, the outgoing data is displayed.

Since it is possible for a field from a Web page to have the same name as a CGI/Environment variable, the letter *E* is inserted in front of the CGI/Environment variables and the letter *V* is inserted in front of the Web page fields.

Testing FCService on a Specified Machine and Port

Use this procedure to verify that the FCService installed is the correct version, running properly, and configured to run on the specified port.

To test FCService:

1. Telnet to the machine and port to test.
2. Press Enter one or more times.

One of the following occurs:

- The message below displays:
`FlashCONNECT <version> invalid message`
where *version* is the FCService version running.
- A connect failed message displays indicating one of these possible problems:
 - FCService is not running.
 - FCService is configured to run on a different port.
 - A firewall is blocking entry to the machine.
- Telnet connects but nothing displays, indicating one of these possible problems:
 - Another service (other than FCService) may be listening on the port.
 - An older FCService version may be listening on the port.

Logging Diagnostic Messages for FCService and FCSocketServer

You can log diagnostic messages to a file using either of the FCService and FCSocketServer commands with the `-Dn` option, where *n* is any combination of these values:

Value for <i>n</i>	Description
0	Logs constructor and deconstructor diagnostic messages.
1	Logs miscellaneous diagnostic messages.
2	Logs input and output diagnostic messages.

For example:

- `FCSocketServer -D12`

Logs FCSocketServer miscellaneous and input/output diagnostic messages.

- `FCService -D0`

Logs FCService constructor and deconstructor diagnostic messages.

- `FCService -D012`

Logs FCService constructor and deconstructor, miscellaneous, and input/output diagnostic messages.

All messages are logged to the `FCdebug000n.txt` file, where *n* is either 0 or 1. For example, when the default file (`FCdebug0000`) reaches its capacity, it is closed and all subsequent messages are logged to `FCdebug0001`. However, if `FCdebug0001` reaches its capacity, it is closed and subsequent messages are logged to `FCdebug0000` (which is cleared of all previous messages).

NOTE— The format of this file changes between releases and is not documented. Customer support may request that you send a copy of this file.

The path for the log file is set as follows:

UNIX	Set to the path contained in the FlashCONNECT <code>FCDEBUG_PATH</code> environment variable. If no path is set in <code>FCDEBUG_PATH</code> , defaults to <code>C:/tmp</code> .
Windows	<p>Set to the first path found in these environment variables, which are checked in the order listed:</p> <ul style="list-style-type: none">• FlashCONNECT <code>FCDEBUG_PATH</code>• Windows <code>TEMP</code>• Windows <code>TMP</code> <p>If no path is set in any of these environment variables, defaults to <code>C:\tmp</code>.</p>

Examining CGI and Form Variables

To examine the cgi and form variables being passed between the browser, Web server and FlashCONNECT, redirect the URL that submits the form from your application to a URL within the cgidisplay application provided with FlashCONNECT. For example:

- Change this URL that submits the form from your application:

<Form Action=/cgi-bin/fccgi.exe?w3exec=*MyApp*>

to this URL that submits the form from the cgidisplay application:

<Form Action=/cgi-bin/fccgi.exe?w3exec=*cgidisplay*>

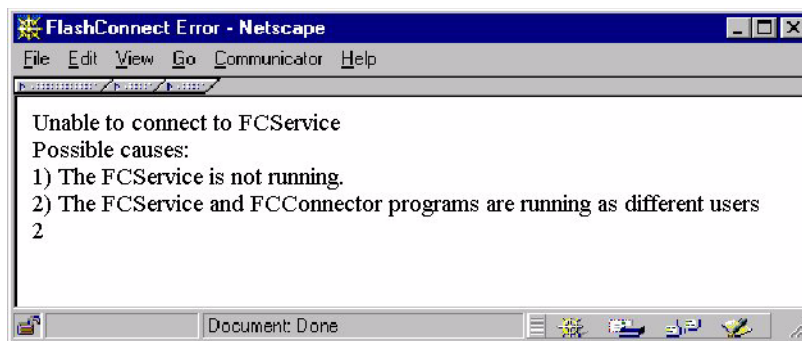
-or-

- Turn on w3Debugging.

Diagnostic Messages Displayed On the Browser

Unable to connect to FCService

If FlashCONNECT is unable to connect to the FCService, possible causes display on the screen. For example:



The number in the lower left of the window (in this case 2) is the most recent Win32 error. Possible solutions are:

Problem	Solution
The FCService is not running.	Use the Services program from the Control Panel to determine if the FlashCONNECT service is running. Start it if it is not running.
The FCService and FCConnector applications are running as different users.	<p>The anonymous Web server user and the FlashCONNECT service must both be able to open the named pipes created by FCService.</p> <ul style="list-style-type: none">• Use the Services program from the Control Panel to determine the name that the FlashCONNECT service runs as.• Use the Internet Service Manager program to determine the Anonymous Logon Username.• If they are not the same, use the Services program from the Control Panel to change the FlashCONNECT service name to match the Anonymous Logon Username.

Determining Release Levels

Use these methods to determine the release level of the FCService, fccgi.exe, and the D3 components.

To determine the release level of FCService or fccgi.exe:

1. Go to the shell prompt.

2. To see the version for:

- FCService, enter:

FCService -V

- FCcgi, enter:

fccgi.exe -V

To determine the release level of the D3 component:

1. Load the FlashCONNECT Main Menu by entering **m** from TCL in the www account.

2. Select the Diagnostic menu.

3. Select View FlashCONNECT version.

or

From TCL, enter:

www-version

Troubleshooting for UNIX Platforms

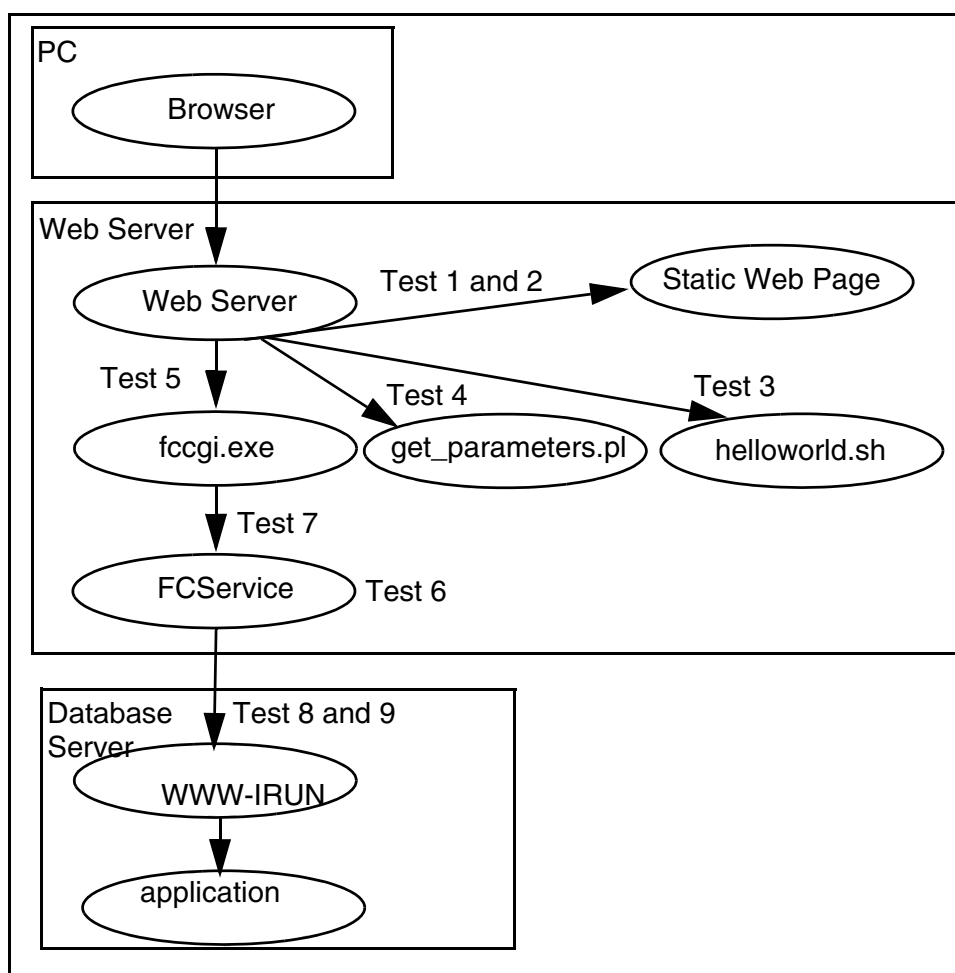
Strategies for troubleshooting problems associated with UNIX platforms are discussed.

For topics not covered here, refer to the [FlashCONNECT Forum](#) listed on the Raining Data Web site.

Troubleshooting for the UNIX Platform

Diagnostics

Assuming the Web server and browser are operating properly and are properly configured, the following processes can be used to diagnose problems while using FlashCONNECT. If a problem occurs, please perform each step in the diagnostic process in order to isolate the problem (see diagram below). Test details are found on the following pages.



Test 1. Verify the browser can access a non-FlashCONNECT HTML page.

Setup:	<ul style="list-style-type: none"> FlashCONNECT does not have to be installed. The Web server must be running. The browser must be able to connect to the Web server.
Process:	Enter the URL to a Web page on your Web site.
Success:	The expected Web page displays.
Failure:	Web server or browser may be misconfigured, not running, or the network connection may be down.
Follow-up:	This problem does not involve FlashCONNECT. Contact your site administrator for assistance.

Test 2. Verify the browser can access the FlashCONNECT static Web pages.

Setup:	<ul style="list-style-type: none"> FlashCONNECT must be installed on your Web server. FlashCONNECT need not be installed on the database server.
Process:	<p>Use a browser and load a static Web page in the w3library directory. The URL will look something like:</p> <p><code>http://yourwebhost/w3library/index.html</code></p> <p>For example:</p> <p><code>http://mywebhost.rainingdata.com/w3library/index.html</code></p>
Success:	FlashCONNECT welcoming page displays.
Failure:	Other than the static Web page, this test does not use FlashCONNECT. Most likely the problem is with the Web site configuration.
Follow-up:	<p>Confirm the following are true:</p> <ul style="list-style-type: none"> The URL is correct. The Web server has read access to the files in the w3library directory.

Test 3. Verify the Web server can run a script in the cgi-bin directory, using sh.

NOTE— This diagnostic tests the ability for the Web server to execute shell scripts. Since FlashCONNECT does not use shell scripts, it is possible for FlashCONNECT to work even if this diagnostic fails.

Setup:	<ul style="list-style-type: none">• FlashCONNECT must be installed on your Web server.• FlashCONNECT need not be installed on the database server.• Shell (/bin/sh) scripts must be executable by the Web server. If they are not, try the diagnostic test, Test 4. Verify the Web server can run a script in the cgi-bin directory, using PERL.• Copy the helloworld.sh program from the w3library directory to the cgi-bin directory on the Web server.
Process:	Use a browser to access the helloworld.sh file. The URL will look something like: <code>http://yourwebhost/path/helloworld.sh</code> For example: <code>http://mywebhost.rainingdata.com/cgi-bin/helloworld.sh</code>
Success:	A Web page containing the message "Hello World" displays.
Failure:	An error message displays.
Follow-up:	Confirm the following are true: <ul style="list-style-type: none">• URL is correct.• Web server has execute access to the files in the cgi-bin directory.• Web server is running on a UNIX system.

Test 4. Verify the Web server can run a script in the cgi-bin directory, using PERL.

NOTE— This diagnostic tests the ability for the Web server to execute PERL scripts. Since FlashCONNECT does not use PERL scripts, it is possible for FlashCONNECT to work even if this diagnostic fails.

Setup:	<ul style="list-style-type: none"> FlashCONNECT must be installed on your Web server. FlashCONNECT need not be installed on the database server. The PERL scripting language must be installed on the Web server. Copy the following applications from the w3library directory to the cgi-bin directory on the Web server. then type: get_parameters.pl, get_parameters.pm.
Process:	<p>Use a browser to access the get_parameters.pl file. The URL will look something like: http://yourwebhost/path/get_parameters.pl For example: http://mywebhost.rainingdata.com/cgi-bin/get_parameters.pl</p>
Success:	A Web page showing the parameters and environmental variables displays. Generally the parameters section is empty. The environmental section is not usually empty.
Failure:	An error message displays.
Follow-up:	<p>Confirm the following are true:</p> <ul style="list-style-type: none"> The URL is correct. PERL exists on the Web server. PERL is accessible by the Web server. The Web server has execute access to the files in the cgi-bin directory.

Test 5. Verify the Web server is using fccgi.exe.

Setup:	<ul style="list-style-type: none">• FlashCONNECT must be installed on your Web server.• FlashCONNECT need not be installed on your database server.• Your Web server must be running.
Process:	<p>Use a browser to access the fccgi.exe program. The URL will look something like:</p> <p><code>http://yourwebhost/path/fccgi.exe</code></p> <p>For example:</p> <p><code>http://mywebhost.rainingdata.com/cgi-bin/fccgi.exe</code></p>
Success:	<p>This message displays:</p> <p>" is an illegal command.</p>
Failure:	<p>A server error occurs.</p>
Follow-up:	<p>Confirm the following are true:</p> <ul style="list-style-type: none">• The URL is correct.• The Web server has access to the cgi-bin directory.• fccgi.exe is executable by the UNIX user used by the Web server.• fccgi.exe is an executable file.• fccgi.exe is in the correct directory. <p>These two diagnostics may help resolve this problem:</p> <ul style="list-style-type: none">• Test 3. Verify the Web server can run a script in the cgi-bin directory, using sh.• Test 4. Verify the Web server can run a script in the cgi-bin directory, using PERL.

Test 6. Verify that the FCService is running.

Setup:	FlashCONNECT must be installed on the Web server.
Process:	<p>As root on the Web server, execute one of the following two lines. Depending upon which type of UNIX is being used, one line will produce a harmless error, the other one will not. Use the one that does not produce an error message.</p> <pre>ps -ef grep FC</pre> <pre>ps -auwx grep FC</pre> <p>One or more lines should display.</p> <ul style="list-style-type: none">• One line contains the text <code>grep FC</code>.• Several lines contain the entire word <code>FCService</code>.

Success:	Several lines contain the entire word <code>FCService</code> .
Failure:	No lines contain the word <code>FCService</code> .
Follow-up:	If no lines display, then verify the <code>FCService</code> file is executable and verify the location of the <code>flashconnect.conf</code> file. Make sure the process used to start the <code>FCService</code> is correct.

Test 7. Confirm the `fccgi.exe` program can access the `FCService` program.

Setup:	<code>FCService</code> must be running on the Web server.
Process:	From the UNIX prompt in the directory that contains <code>FCdebug</code> , execute: <code>./FCdebug -h</code>
Success:	<code>FCService</code> configuration displays.
Failure:	An error message displays.
Follow-up:	See the following sections: <ul style="list-style-type: none"> • Using the FlashCONNECT Server Debug Program • Diagnostic Messages Displayed On the Browser

Test 8. Check if the `FCService` and D3 server are connecting with each other. Use either of these two methods as a valid diagnostic.

a. Method One

Setup:	<ul style="list-style-type: none"> • FlashCONNECT must be installed and running on the Web server. • FlashCONNECT must be installed and running on the database server. • Determine the first few characters of the database server's host name.
Process:	<p>As root on the Web server, enter:</p> <p><code>netstat -a grep hostname</code></p> <p>For example:</p> <p><code>netstat -a grep eng</code></p> <ul style="list-style-type: none"> • One or two lines for each connection between the <code>FCService</code> and the database server should display.

Success:	Status <code>ESTABLISHED</code> displays on line one for each connection. The status <code>LISTEN</code> displays on the second line, if it exists.
Failure:	There is not an entry with the status <code>ESTABLISHED</code> for each connection to database.
Follow-up:	<ul style="list-style-type: none">• Verify that <code>FCService</code> is running.• Verify <code>FlashCONNECT</code> was started on the database server.• Shut down and restart both the Web and database components of <code>FlashCONNECT</code>.

b. Method Two

Setup:	<ul style="list-style-type: none">• <code>FlashCONNECT</code> must be installed and running on the Web server.• <code>FlashCONNECT</code> must be installed and running on the database server.
Process:	From the UNIX prompt in the directory that contains <code>FCdebug</code> , enter: <code>./FCdebug -t</code>
Success:	The appropriate server pools are listed in the Host Name column.
Failure:	The appropriate server pools are not listed in the Host Name column.
Follow-up:	<ul style="list-style-type: none">• Check that <code>FCService</code> is running.• Shut down and restart the D3 <code>FlashCONNECT</code> component.• Shut down and restart <code>FCService</code>.• Shut down and restart database.• Shut down and restart UNIX.

Test 9. Check that the FCService can access the database server.

Setup:	<ul style="list-style-type: none"> FlashCONNECT must be installed and running on the Web server. FlashCONNECT must be installed and running on the database server. Copy the <code>fccgi.to.d3.test.sh</code> script to the Web server's <code>cgi-bin</code> directory. As needed, modify the script to correct the location of the <code>fccgi.exe</code>, <code>grep</code>, and <code>echo</code> commands. The FlashCONNECT demonstration applications must be installed on the database server (default configuration).
Process:	From the UNIX shell on the Web server, in the <code>cgi-bin</code> directory, run the <code>fccgi.to.d3.test.sh</code> script.
Success:	The message <code>connection ok</code> displays.
Failure:	The message <code>connection failed</code> or any other error messages display.
Follow-up:	<ul style="list-style-type: none"> Confirm that the Boo program is in the database server in the <code>www bp</code> file. Check that FCService is running. Check that database FlashCONNECT connections are established to the Web server using <code>WWW-STATUS</code>. See Using WWW-STATUS. Use <code>fcdebug -t</code> to confirm FCService is connected to D3 server. Shut down and restart FCService. Shut down and restart database. Shut down and restart UNIX.

Using the fccgi.pl trace Utility

The `fccgi.pl` trace utility is a FlashCONNECT diagnostic tool for tracing data passage between a FlashCONNECT server and a Web server. If you suspect that data is not being properly passed between `fccgi.exe` and the Web server, use `fccgi.pl` to trace the passed information.

Troubleshooting for Windows Platforms

Strategies for troubleshooting problems associated with Windows platforms and provides a description of Event Log error messages are discussed.

For topics not covered here, refer to the [FlashCONNECT Forum](#) listed on the Raining Data Web site.

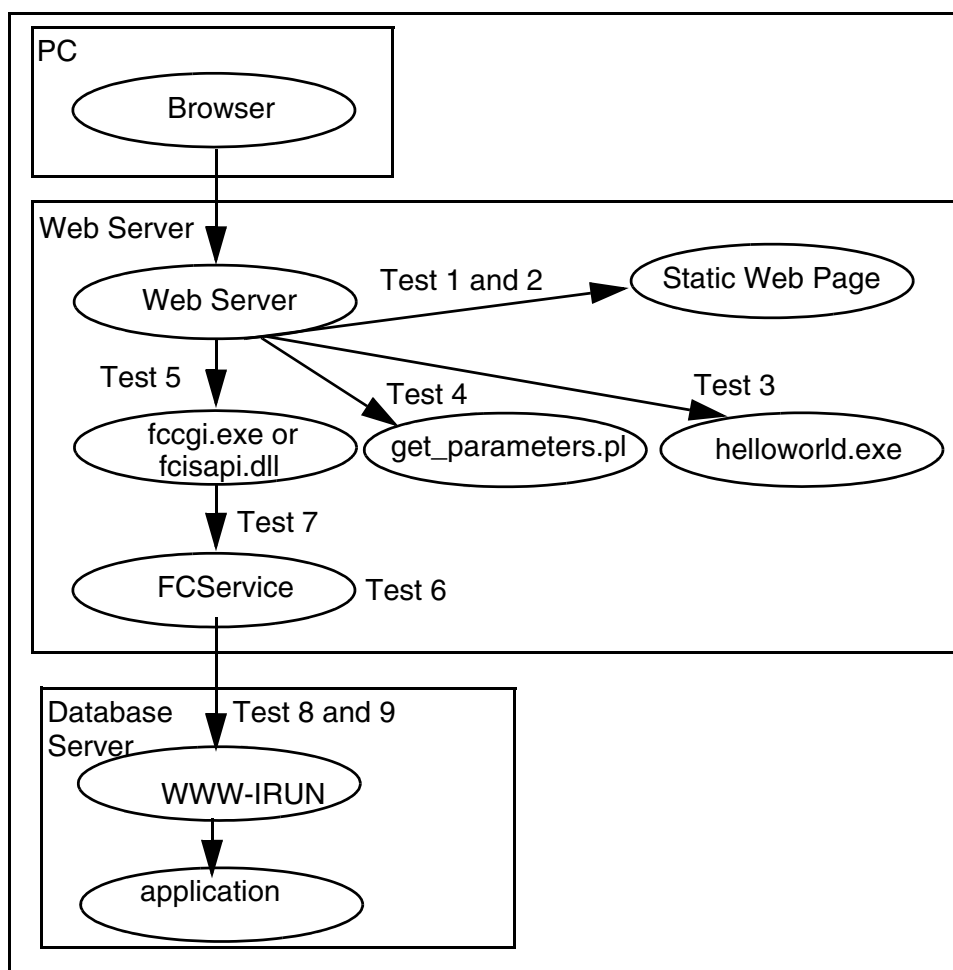
Troubleshooting for Windows

This section documents:

- Diagnostic steps to isolate problems.
- Diagnostic messages shown in the browser.
- Event log entries written by FlashCONNECT.

Diagnostics

Assuming the Web server and browser are operating properly and are properly configured, the following processes can be used to diagnose problems while using FlashCONNECT. If a problem occurs, perform each step in the diagnostic process in order to isolate the problem. Test details are found on the following pages.



Test 1. Verify the browser can access a non-FlashCONNECT HTML page.

Setup:	<ul style="list-style-type: none"> FlashCONNECT does not have to be installed. The Web server must be running. The browser must be able to connect to the Web server.
Process:	Enter the URL to a Web page on your Web site.
Success:	Expected Web page displays.
Failure:	Web server or browser may be misconfigured, not running, or the network connection may be down.
Follow-up:	This problem does not involve FlashCONNECT. Contact your site administrator for assistance.

Test 2. Verify the browser can access the FlashCONNECT static Web pages.

Setup:	<ul style="list-style-type: none"> FlashCONNECT demonstration applications (WebContent/w3library) <i>must</i> be installed on your Web server. FlashCONNECT need not be installed on the database server.
Process:	<p>Use a browser and load a static Web page in the w3library directory. The URL will look something like:</p> <p><code>http://yourwebhost/w3library/index.html</code></p> <p>For example:</p> <p><code>http://mywebhost.rainingdata.com/w3library/index.html</code></p>
Success:	The FlashCONNECT welcoming page displays.
Failure:	Other than the static Web page, this test does not use FlashCONNECT. Most likely the problem is with the Web site configuration.
Follow-up:	<p>Confirm the following are true:</p> <ul style="list-style-type: none"> The URL is correct. The Web server has read access to the files in the w3library directory. <p>This problem does not involve FlashCONNECT. Contact your site administrator for assistance.</p>

Test 3. Verify the Web server can run a script in the cgi-bin directory.

Setup:	<ul style="list-style-type: none">• FlashCONNECT must be installed on the Web server.• FlashCONNECT need not be installed on the database server.• FlashCONNECT does not have to be running on the Web server nor D3 server.• Copy the program helloworld.exe from the WebContent directory to the Web server's cgi-bin directory where fccgi.exe is located.
Process:	<p>Use a browser to access the helloworld.exe program. The URL will look similar to the following:</p> <p><code>http://yourwebhost/path/helloworld.exe</code></p> <p>For example:</p> <p><code>http://mywebhost.rainingdata.com/cgi-bin/helloworld.exe</code></p>
Success:	A Web page containing the message "Hello World!" displays.
Failure:	An error message displays.
Follow-up:	<p>Confirm the following are true:</p> <ul style="list-style-type: none">• The URL is correct.• The Web server has execute access to the directory.• Windows file and directory permissions allow execution. <p>Perform these actions:</p> <ul style="list-style-type: none">• Shut down and restart the Web server.• Shut down and restart Windows. <p>This problem does not involve FlashCONNECT. Contact your site administrator for assistance.</p>

Test 4. Verify the Web server can run a script in the cgi-bin directory, using PERL.

NOTE—This diagnostic tests the ability for the Web server to execute PERL scripts. Since FlashCONNECT does not use PERL scripts, it is possible for FlashCONNECT to work even if this diagnostic fails.

Setup:	<ul style="list-style-type: none"> FlashCONNECT must be installed on your Web server. FlashCONNECT need not be installed on the database server. The PERL scripting language must be installed on the Web server. Copy the following applications from the w3library directory to the cgi-bin directory on the Web server: get_parameters.pl, get_parameters.pm
Process:	<p>Use a browser to access the get_parameters.pl file. The URL will look something like: http://yourwebhost/path/get_parameters.pl For example: http://mywebhost.rainingdata.com/cgi-bin/get_parameters.pl</p>
Success:	A Web page showing the parameters and environmental variables displays. Generally the parameters section is empty and the environmental section is not usually empty.
Failure:	An error message displays.
Follow-up:	<p>Confirm the following are true:</p> <ul style="list-style-type: none"> The URL is correct. PERL exists on the Web server. PERL is accessible by the Web server. The Web server has execute access to the files in the cgi-bin directory. <p>This problem does not involve FlashCONNECT. Contact your site administrator for assistance.</p>

Test 5. Verify the Web server is using fccgi.exe.

Setup:	<ul style="list-style-type: none">• FlashCONNECT must be installed on your Web server.• FlashCONNECT need not be installed on your database server.• Your Web server must be running.
Process:	<p>Use a browser to access the fccgi.exe program. The URL will look something like:</p> <p><code>http://yourwebhost/path/fccgi.exe</code></p> <p>For example:</p> <p><code>http://mywebhost.rainingdata.com/cgi-bin/fccgi.exe</code></p>
Success:	<p>One or both of these messages display to indicate success:</p> <ul style="list-style-type: none">• ERROR - Missing required elements in form• QUERYSTRING must be set for CGI
Failure:	<p>A server error occurs.</p>
Follow-up:	<p>Confirm the following are true:</p> <ul style="list-style-type: none">• The URL is correct.• The Web server has access to the directory containing fccgi.exe.• fccgi.exe is executable by the user used by the Web server.• fccgi.exe is an executable file.• fccgi.exe is in the correct directory.• Files in the directory are executable by the Web server. <p>These two diagnostics may help resolve this problem:</p> <ul style="list-style-type: none">• Test 3. Verify the Web server can run a script in the cgi-bin directory.• Test 4. Verify the Web server can run a script in the cgi-bin directory, using PERL.

NOTE—This diagnostic is not designed for use with fcisapi.dll. If it is used in that manner, the results are meaningless.

Test 6. Verify that the FCService is running.

Setup:	FlashCONNECT must be installed on the Web server.
Process:	From the command prompt, enter these commands: <code>cd path</code> <code>fcdebug -h</code> where: <i>path</i> is the location where FlashCONNECT is installed.
Success:	FlashCONNECT header information displays. A description of the header information is located in Using the FlashCONNECT Server Debug Program .
Failure:	FlashCONNECT not running message displays.
Follow-up:	The following options are available from the Services program (accessed from the Control Panel): <ul style="list-style-type: none"> • If the FCService is not running, start FCService. • If the FCService is already running, stop FCService. Restart FCService again. • Stop the Web server and FCService. Restart the Web server and FCService. • Stop the database, the Web server and FCService. Restart the database, the Web server and FCService.

Test 7. Confirm the fccgi.exe program can access the FCService program.

Setup:	FCService must be running on the Web server.
Process:	From the MS-DOS command prompt in the directory that contains <code>fcdebug</code> , enter: <code>fcdebug -h</code>
Success:	FCService configuration displays.
Failure:	An error message displays.
Follow-up:	See the following sections: <ul style="list-style-type: none"> • Using the FlashCONNECT Server Debug Program • Diagnostic Messages Displayed On the Browser

Test 8. Verify that FCService and the database server are connecting with each other. Use either of these two methods as a valid diagnostic.

a. Method One

Setup:	<ul style="list-style-type: none">• FlashCONNECT must be installed and running on the Web server.• FlashCONNECT must be installed and running on the database server.
Process:	From the MS-DOS command prompt in the directory that contains fcdebug, enter: fcdebug -t
Success:	Appropriate server pools are listed in the Host Name column.
Failure:	Appropriate server pools are not listed in the Host Name column.
Follow-up:	<ul style="list-style-type: none">• Check that FCService is running.• Shut down and restart the D3 FlashCONNECT component.• Shut down and restart FCService.• Shut down and restart database.• Shut down and restart Windows.

b. Method Two

Setup:	<ul style="list-style-type: none">• FlashCONNECT must be installed and running on the Web server.• FlashCONNECT must be installed and running on the database server.• Determine the first few characters of the database server's host name.
Process:	From the MS-DOS prompt on the Web server, enter: netstat -a grep hostname For example: netstat -a grep eng One or two lines for each connection between the FCService and database server should display.
Success:	Status ESTABLISHED displays on line one for each connection. The status LISTEN displays on the second line, if it exists.
Failure:	There is not an entry with the status ESTABLISHED for each connection to database.

Follow-up:	<ul style="list-style-type: none"> • Use the Task Manager to verify there is only one copy of FCService running. • Verify FlashCONNECT was started on the database server. • Shut down and restart both the Web and database components of FlashCONNECT.
------------	---

Test 9. Check that the fccgi.exe program can access the database server.

Setup:	<ul style="list-style-type: none"> • FlashCONNECT must be installed and running on the Web server. • FlashCONNECT must be installed and running on the database server. • The FlashCONNECT demonstration program must be installed on the database server. • Copy the program fccgi.to.d3.test.bat from the FlashCONNECT WebContent directory to the Web server's script (cgi-bin) directory where fccgi.exe is located.
Process:	From the MS-DOS command prompt in the directory that contains both fccgi.exe and fccgi.to.d3.test.bat, enter: fccgi.to.d3.test
Success:	Resulting message contains the text <code>connection OK</code> .
Failure:	Resulting message contains the text <code>connection failed</code> .
Follow-up:	<ul style="list-style-type: none"> • Confirm that the Boo program is in the database server in the www bp file. • Check that FCService is running. • Check that database FlashCONNECT connections are established to the Web server using <code>WWW-STATUS</code>. See Using WWW-STATUS. • Use <code>fcdebug -t</code> to confirm FCService is connected to database server. • Shut down and restart FCService. • Shut down and restart database. • Shut down and restart Windows.

Using the fccgi.pl trace Utility

The fccgi.pl trace utility is a FlashCONNECT diagnostic tool for tracing data passage between a FlashCONNECT server and a Web server. If you suspect that data is not being properly passed between fccgi.exe and the Web server, use fccgi.pl to trace the passed information.

FlashCONNECT Event Log Entries

FCService writes messages to the Windows Event Log. Some messages are informative and others may indicate that an error occurred. This section describes each message written to the Event Log by FCService.

MSG_STARTED

Type	Informative.
Meaning	FlashCONNECT Service started successfully.
When	Displays when the FlashCONNECT service starts.
Action	None.

MSG_STOPPED

Type	Informative.
Meaning	FlashCONNECT Service stopped successfully.
When	FlashCONNECT service stops intentionally.
Action	None.

MSG_RUNNING

Type	Warning.
Meaning	FlashCONNECT Service is running. It cannot start again.
When	Displays when an attempt is made to start the FlashCONNECT service when it is already running.
Action	Do not attempt to run more than one copy of FCService at a time.

MSG_NOSLOTS

Type	Warning.
Meaning	FCService shared memory has no space available for additional connections from database.
When	Requests for additional connections from database servers to this FCService exceed the number that FCService can support.
Action	<ul style="list-style-type: none">• Configure FCService to support additional connections by increasing the D3ServerSlots parameter in the configuration.• Restart the FCService.

MSG_NOSHM

Type	Warning.
------	----------

Meaning Unable to attach to the shared memory segment.
 When An attempt was made to use FCService before it was running.
 Action Verify that FCService is running.

MSG_LOSTCOOKIES

Type Warning.
 Meaning FCService lost a cookie.
 When A user returns to a form that has already timed out.
 Action If this message is received often, check the timeout values in your application. They may be too short.

MSG_SPIN

Type Warning.
 Meaning FlashCONNECT Connector (FCCGI.exe) could not locate an open database server on its first attempt, but was subsequently able to find one. The message displays the name of the server pool in which the connection is made.
 When An indication that not enough database server phantoms exist to service all of the traffic. Any increase in traffic will cause browser errors.
 Action Start more FlashCONNECT connections for this server pool.

MSG_NOSERVERS

Type Warning.
 Meaning FlashCONNECT Connector (FCcgi.exe) could not locate an open database server phantom. The message displays the name of the server pool in which the attempt to connect was made.
 When An indication that not enough database server phantoms exist to service all of the traffic. The browser received an error message.
 Action Increase the number of database server phantoms.

MSG_EXCEPTION

Type Informative/Warning.
 Meaning The following internal error occurred: <message>.
 When Catch-all messages that display when no other messages apply.
 Action Undefined.

MSG_Exception Messages

We describe MSG_Exception error messages.

For topics not covered here, refer to the [FlashCONNECT Forum](#) listed on the Raining Data Web site.

MSG_Exception: Event Handler Messages

FlashCONNECT provides the MSG_Exception messages that display in either the Windows Event Handler or in the UNIX Log file. The following describes the different message types, their meaning, the severity and what action to take.

There are two types of severity:

Page will fail	Processing of the submitted page will fail. FlashCONNECT can continue processing valid requests.
FCService will fail	FCService will stop.

400, CONTENT_LENGTH must be set for CGI

Meaning: Web server did not set the CONTENT_LENGTH parameter. It must be specified for CGI.

Severity: Page will fail.

Action: Contact the Web server provider.

400, CONTENT_TYPE!=application/x-www-form-urlencoded

Meaning: FlashCONNECT expects the CONTENT_TYPE to be set to application/x-www-form-urlencoded. Other encodings are not supported.

Severity: Page will fail.

Action: Change the encoding on the Web page being submitted.

400, CONTENT_TYPE must be set for CGI

Meaning: Web server did not set the CONTENT_TYPE parameter. It must be specified for CGI.

Severity: Page will fail.

Action: Contact the Web server provider.

400, ERROR: Unknown method

Meaning: Method must be specified as either GET or POST. The form being submitted was not submitted as a POST or GET.

Severity: Page will fail.

Action: Correct the form being submitted.

400, pecb, ERROR: ReadClient failed

Meaning: FCisapi failure processing very large input value.

Severity: Page will fail.

Action: Internal error. Contact Raining Data Customer Support.

400, pecb, ERROR: Unknown method

Meaning: Method must be specified as either GET or POST. The form being submitted was not submitted as a POST or GET.

Severity: Page will fail.

Action: Correct the form being submitted.

400, QUERY_STRING must be set for CGI

Meaning: When REQUEST_METHOD is set to GET, the QUERY_STRING parameter must be set.

Severity: Page will fail.

Action: Correct the form being submitted.

400, REQUEST_METHOD must be set for CGI

Meaning: A REQUEST_METHOD (either GET or POST) must be specified. The form being submitted was not submitted as a POST or GET.

Severity: Page will fail.

Action: Correct the form being submitted.

accept() failed, errorcode

Meaning: FCSservice encountered an invalid socket while attempting to accept a new connection.

Severity: Page will fail, FCSservice may become unstable.

Action: Check errorcode and resolve the underlying problem.

bind() failed

Meaning: Unable to bind a socket to an address.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

bind(): only AF_INET supported

Meaning: The bind failed. FCSservice only supports AF_INET for binding.

Severity: Internal error. FCSservice will fail.

Action: Contact Raining Data Customer Support.

bind(): Unable to get host address

Meaning: Internal error. GetHostAddress returned a 0 (zero).

Severity: Page will fail. FCSservice may become unstable.

Action: Contact Raining Data Customer Support.

Database tried to connect with the incorrect protocol version

Meaning: Protocol versions between FCSservice and database are not synchronized.

Severity: FCSservice will fail.

Action: Reload one version or the other to synchronize the versions.

EnvVar() ERROR_INVALID_PARAMETER

Meaning: There was a problem getting the server variables.

Severity: Page will not be returned.

Action: Check the Web server configuration. FCisapi (Windows) only.

EnvVar() ERROR_MORE_DATA

Meaning: There was a problem getting the server variables.

Severity: Page will not be returned.

Action: Check the Web server configuration. FCisapi (Windows) only.

Failed to create semaphore set, errorcode

Meaning: CreateSemaphore returned a null pointer.

Severity: Page will fail. FCService will not process pages.

Action: Check errorcode and resolve underlying problem.

Failed to create semaphore set (new), errorcode

Meaning: Out of memory when attempting to get an array of semaphore handles.

Severity: Page will fail. FCService will not process pages.

Action: Check errorcode and resolve underlying problem.

Failed to open semaphore, errorcode

Meaning: Unable to open a new or existing semaphore.

Severity: Page will fail. FCService will not process pages.

Action: Check errorcode and resolve underlying problem.

Failure while looking up server during send of a pulse. Try again in a few moments.

Meaning: Could not find a server when sending heartbeat pulse. Connections may time out unexpectedly.

Severity: Page will fail.

Action: None.

FCisapi.dll: Redirect string too long, FCCBLocSize

Meaning: Buffer overrun in FCisapi (Windows) only.

Severity: Page will fail.

Action: Verify expected size of returned page.

FCServiceStore::D3SeverNew() s.consume

Meaning: Unable to get a semaphore for a new worker thread.

Severity: Page will fail. FCService will not process pages.

Action: Verify that a process can have as many threads as are being started.

listen() failed

Meaning: listen returned a non-zero value for a w3socket.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

Protocol Version Mismatch, ProtocolVersion

Meaning: Internal protocol versions of FCService and the FlashCONNECT database component (www account) do not match.

Severity: FCService will fail.

Action: Reload one version or the other to synchronize the versions.

Semaphore consumption failed, errorcode

Meaning: Failure attempting to consume the semaphore.

Severity: Page will fail.

Action: Check errorcode.

Semaphore consumption failed, errorcode

Meaning: WaitForSingleObject failed for semaphore.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

Semaphore get-value failed, errorcode

Meaning: Bad status returned while attempting to get the value of a semaphore.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

Semaphore production failed, errorcode

Meaning: Unable to produce the required semaphore count.

Severity: Page will fail. FCSservice will not process pages.

Action: Check errorcode and resolve underlying problem.

Sorry, no FlashCONNECT servers are available at this time. Try again in a few moments.

Meaning: Insufficient resources.

Severity: Page will fail.

Action: Start more database FlashCONNECT servers to support this server pool.

That connection is invalid or has timed out.

Meaning: w3SessionId is invalid because the user waited too long to respond.

Severity: Page will fail.

Action: None.

Unable to connect to FCSservice

Meaning: No connections will be made. Possible causes are:

- FCSservice is not running.
- FCSservice and FCconnector applications are running as different users.
- FCSservice and FCconnector versions are not compatible.

Severity: Pages will fail.

Action: Verify that FCSservice is running properly and versions are correct.

w3Config::create() Unable to create db, errorcode

- Meaning: Failure to open or create the configuration database (registry problems under Windows).
- Severity: FCSservice will fail.
- Action: Check errorcode and resolve underlying problem. Verify presence and permissions of flashconnect.conf (UNIX) or registry entries (Windows).

w3Config::create() Use an unopened object

- Meaning: Configuration Internal error. Configuration database appears to already be open.
- Severity: FCSservice will fail.
- Action: Contact Raining Data Customer Support.

w3Config::get() Unable to read key, errorcode

- Meaning: Unable to read a key from the registry.
- Severity: FCSservice will fail.
- Action: Check errorcode and resolve underlying problem. Possible reinstall.

w3Config::open() Invalid hive

- Meaning: Bad hive name while opening the registry. Windows only.
- Severity: FCSservice will fail.
- Action: Fix/correct registry problem.

w3Config::open() Invalid path string

- Meaning: Internal error. Bad path while opening the registry. Windows only.
- Severity: FCSservice will fail.
- Action: Contact Raining Data Customer Support.

w3Config::open() Unable to open db, errorcode

Meaning: Unable to open the configuration file under UNIX.
Severity: FCService will fail.
Action: Check errorcode and resolve underlying problem. Check permissions on the configuration file.

w3Config::open() Unable to open key, errorcode

Meaning: Unable to open a key value in the registry. Windows only.
Severity: FCService will fail.
Action: Check errorcode. Ensure FlashCONNECT is properly installed.

w3Config::wait() Unable to get status of file, errorcode

Meaning: Problem while attempting to see if the configuration file has changed under UNIX.
Severity: FCService will fail.
Action: Check errorcode. Verify permissions on the configuration file.

w3Config::wait() Unable to wait for db change, errorcode

Meaning: Problem while checking the registry for changes. Windows only.
Severity: FCService will fail.
Action: Check errorcode and resolve underlying problem. Ensure FlashCONNECT is properly installed.

w3Connection::OnAccept() Override this function in your derived class.

Meaning: Internal error.
Severity: Internal compiler error.
Action: Contact Raining Data Customer Support.

w3Connection::OnClose() Override this function in your derived class.

Meaning: Internal error.

Severity: Internal compiler error.

Action: Contact Raining Data Customer Support.

w3Connection::OnConnect() Override this function in your derived class.

Meaning: Internal error.

Severity: Internal compiler error.

Action: Contact Raining Data Customer Support.

w3Connection::OnRead() Override this function in your derived class.

Meaning: Internal error.

Severity: Internal compiler error.

Action: Contact Raining Data Customer Support.

w3Connection::OnWrite() Override this function in your derived class.

Meaning: Internal error.

Severity: Internal compiler error.

Action: Contact Raining Data Customer Support.

w3ErrLog::ActuallyOpen() Attempting to write error without opening error log, errorcode

Meaning: Internal error.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3ErrLog::ActuallyOpen() RegisterEventSource() failed, errorcode

Meaning: Failure opening the error log.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::FinishAccept(): accept(), errorcode

Meaning: Unable to finish the accept on the pipe socket. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::make(const char *n): bind(), errorcode

Meaning: Failure to bind while making a socket for a named pipe. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::make(const char *n): CreateEvent(), errorcode

Meaning: Failure to create an event while making a named pipe. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::make(const char *n): CreateNamedPipe(), errorcode

Meaning: Failure to create a named pipe. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::make(const char *n): InitializeSecurityDescriptor(), errorcode

Meaning: Failure to initialize the security descriptor for a new pipe. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::make(const char *n): listen() ,errorcode

Meaning: Failure to listen while making a socket for a named pipe. UNIX only.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::make(const char *n): malloc()

Meaning: Out of memory for a security descriptor while making a named pipe.

Severity: FCSservice will fail.

Action: Check memory usage.

**w3Pipe::make(const char *n): SetSecurityDescriptorDacl(),
errorcode**

Meaning: Failure to set security descriptor DACL while making a named pipe.
Windows only.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::StartAccept(): ConnectNamedPipe()

Meaning: Unable to connect to the named pipe under Windows. Windows only.

Severity: FCSservice will fail.

Action: Check system resources.

w3Pipe::StartAccept(): Unexpected error, errorcode

Meaning: Unexpected errorcode while trying to connect to named pipe under
Windows.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::StartConnect(): connect(), errorcode

Meaning: Failure during socket connection under UNIX.
Severity: FCSservice will fail.
Action: Check errorcode and resolve underlying problem.

w3Pipe::StartConnect(): CreateEvent()

Meaning: Failure to create an event during socket connection start-up under Windows.
Severity: FCSservice will fail.
Action: Check errorcode and resolve underlying problem.

w3Pipe::StartConnect(): CreateFile(), errorcode

Meaning: Unable to create a pipe file under Windows.
Severity: FCSservice will fail.
Action: Check errorcode and resolve underlying problem.

w3Pipe::StartRead() buf!=NULL, bufsz<1

Meaning: Internal error trying to create a second buffer during read from pipe!
Severity: FCSservice will fail.
Action: Contact Raining Data Customer Support.

w3Pipe::StartRead() ERROR_PIPE_NOT_CONNECTED, possible Flash/NonFlash compile mismatch", errorcode

Meaning: **For D3:** It's likely that the application has aborted. This may be due to Flash and non-Flash compiled modules being mixed.
For mvBase: It's likely that the application has aborted.
Severity: Page will fail.
Action: **For D3:** Ensure that all applications are Flash-compiled and running properly.
For mvBase: Ensure that all applications are running properly.

w3Pipe::StartRead() Read already in progress

Meaning: Internal error trying to read while in the middle of a read.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

w3Pipe::StartRead() Unexpected error, errorcode

Meaning: Unexpected status from ReadFile.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::StartWrite() buf==NULL

Meaning: Internal error trying to write a null buffer.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

w3Pipe::StartWrite() bufisz < 1

Meaning: Internal error trying to write a buffer with a negative size.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

w3Pipe::StartWrite() Unexpected error, errorcode

Meaning: Writing to the pipe returned an unexpected error.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

w3Pipe::StartWrite() wrt in prgrs

Meaning: Internal error trying to write while a write is in progress.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

w3Selector::insert() out of memory

Meaning: Out of memory while trying to insert a w3connection.

Severity: FCService will fail.

Action: Check memory availability.

w3Selector::run() select() unknown error, errorcode

Meaning: Unexpected error returned from select while looping of selector. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Selector::run() WaitForMultipleObjects() unknown error

Meaning: Unexpected error returned from WaitForMultipleObjects while looping of selector. Windows only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Selector::run() WaitForMultipleObjects() WAIT_FAILED, errorcode

Meaning: Got WAIT_FAILED error status returned from WaitForMultipleObjects while looping of selector. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Selector::run() WaitForMultipleObjects() WAIT_TIMEOUT

Meaning: Got WAIT_TIMEOUT error status returned from WaitForMultipleObjects while looping of selector. Windows only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Semaphore::consume() - semid <0

Meaning: Trying to consume an illegal semaphore ID. UNIX only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Semaphore::consume() - semnum <0

Meaning: Trying to consume an illegal semaphore number. UNIX only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Semaphore::consume() - timeout <0

Meaning: Invalid timeout while trying to consume a semaphore. UNIX only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Semaphore::create() new failed

Meaning: Out of memory trying to create new semaphores.

Severity: FCService will fail.

Action: Check memory usage.

w3Semaphore::create() semget() failed, errorcode

Meaning: Failure while trying to get a new semaphore.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Semaphore::create(): InitializeSecurityDescriptor(), errorcode

Meaning: Failure of InitializeSecurityDescriptor while trying to get a new semaphore. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Semaphore::create(): malloc()

Meaning: Out of memory while trying to build a new semaphore. Windows only.

Severity: FCService will fail.

Action: Check memory usage.

w3Semaphore::create(): SetSecurityDescriptorDacl(), errorcode

Meaning: Failure of SetSecurityDescriptorDacl while trying to build a new semaphore. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Semaphore::produce() - semid <0

Meaning: Bad semaphore ID while trying to produce new semaphores. UNIX only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Semaphore::produce() - semnum <0

Meaning: Bad semaphore number while trying to produce new semaphores. UNIX only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Semaphore::value() - semid <0)

Meaning: Bad semaphore ID while trying to return the value of a semaphore. UNIX only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Semaphore::value() - semnum <0)

Meaning: Bad semaphore number while trying to return the value of a semaphore. UNIX only.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Semaphore::value() semctl() failed, errorcode

Meaning: Failure of semctl while trying to return the value of a semaphore. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::~w3SharedMemory() shmctl(IPC_RMID), errorcode

Meaning: Shared memory control failed during shared memory destruction.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::~w3SharedMemory() shmdt(), errorcode

Meaning: Shared memory failure during shared memory destruction.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::~w3SharedMemory() UnmapViewOfFileEx(), errorcode

Meaning: Failure to unmap the view of the file during shared memory destruction.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

**w3SharedMemory::create() CreateFileMapping() (exists),
errorcode**

Meaning: Shared memory segment already exists. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem. Verify only one copy of FCService is running.

w3SharedMemory::create() CreateFileMapping(), errorcode

Meaning: Failure to create the shared memory segment. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::create() MapViewOfFileEx(), errorcode

Meaning: Failure to map the shared memory into the address space. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::create() shmat(), errorcode

Meaning: Failure to map the shared memory segment. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::create() shmget(), errorcode

Meaning: Failure to get the shared memory segment. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

**w3SharedMemory::Create(): InitializeSecurityDescriptor(),
errorcode**

Meaning: Problems setting up security descriptor for the shared memory.
Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::Create(): malloc()

Meaning: Out of memory while creating shared memory segment.

Severity: FCService will fail.

Action: Check system resources.

**w3SharedMemory::Create(): SetSecurityDescriptorDacl(),
errorcode**

Meaning: Problems adding a NULL descriptor ACL to the security descriptor.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::open() MapViewOfFileEx(), errorcode

Meaning: Failure trying to map the shared memory. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::open() OpenFileMapping(), errorcode

Meaning: Failure trying to map the shared memory. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::open() shmat(), errorcode

Meaning: Failure to map the shared memory segment. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::open() shmctl(), errorcode

Meaning: Failure to get the size of the shared memory segment. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3SharedMemory::open() shmget(), errorcode

Meaning: Failure to re-open the shared memory segment to the correct size. UNIX only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Socket::FinishRead() You must call StartRead() first

Meaning: Internal error.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Socket::FinishWrite() send() error, errorcode

Meaning: Socket error on send trying to complete a write.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Socket::make() socket()

Meaning: Invalid value while trying to make a socket.

Severity: FCService will fail.

Action: Check system socket configuration.

w3Socket::select() CreateEvent(), errorcode

Meaning: Failure trying to set up a socket for a selector object. Windows only.

Severity: FCService will fail.

Action: Check errorcode.

w3Socket::select() WSAEventSelect(), errorcode

Meaning: Failure trying to attach an event to a socket for a selector object. Windows only.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Socket::StartConnect() Unimplemented

Meaning: Internal error.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Socket::StartRead() buf!=NULL, bufsz<1

Meaning: Internal error. Buffer is already allocated.

Severity: FCService will fail.

Action: Contact Raining Data Customer Support.

w3Socket::StartRead() failure to allocate buf, errcode

Meaning: Internal error or out of memory.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Socket::StartRead() Read already in progress

Meaning: Internal error. Starting a read while already in progress.

Severity: FCService will fail.

Action: Check errorcode and resolve underlying problem.

w3Socket::StartWrite() buf==NULL

Meaning: Internal error. Trying to write a null buffer.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

w3Socket::StartWrite() bufsz<1

Meaning: Internal error. Trying to write a buffer of negative size.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

w3Socket::StartWrite() write already in progress

Meaning: Internal error. Starting a write while a write is already in progress.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

w3Socket::w3Socket(SOCKET, af)

Meaning: Internal error. Address family of socket illegal.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

w3Thread::kill(), errorcode

Meaning: Failure trying to terminate a thread. UNIX only.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

w3Thread::w3Thread(): CreateThread(), errorcode

Meaning: Failure to create a thread structure. Windows only.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

w3Thread::w3Thread(): fork(), errorcode

Meaning: Failure to start the new thread. UNIX only.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

w3Thread::wait(), errorcode

Meaning: Failure to wait on a thread.

Severity: FCSservice will fail.

Action: Check errorcode and resolve underlying problem.

Winsock version 2.0 required

Meaning: Failure while trying to init with WSStartup under Windows.

Severity: FCSservice will fail.

Action: Install proper release of Winsock.

WorkerSocket::OnRead() Bad state, errorcode

Meaning: Internal error. Socket has an illegal state.

Severity: FCSservice will fail.

Action: Contact Raining Data Customer Support.

WorkerSocket::OnRead() w3ServerPool name too long

Meaning: Specified w3ServerPool name is longer than 15 characters.

Severity: FCSservice will fail.

Action: Shorten the w3ServerPool name (in the w3Config file). The limit is 15 characters.

Error Messages

We define common error messages that you might encounter.

For topics not covered here, refer to the [FlashCONNECT Forum](#) listed on the Raining Data Web site.

UNIX Platform Error Messages

Server Error

If accessing a dynamic application, such as the w3Admin, and the browser reports a Server Error, the FlashCONNECT servers have probably not been set up correctly. See the documentation for `WWW-STATUS`, `WWW-START` and `WWW-STOP`.

Unknown line in configuration file: xxxx

When the FCService starts, it reads the flashconnect.conf file. At this time, if an unexpected line is found, the above message is printed.

To recover, correct or remove the unexpected line and restart the FCService.

ERROR - Missing required elements in form

This message displays if the URL starting a FlashCONNECT application is missing either a w3Exec parameter or a w3SessionId parameter. This also happens if the calls to w3HtmlFormBeg or w3HtmlFormEnd are missed.

This URL example shows the error:

`http://w3lib.rainingdata.com/cgi-bin/fccgi.exe?w3ClosingPage=!url`

This URL example shows the correction:

`http://w3lib.rainingdata.com/cgi-bin/fccgi.exe?w3exec=app&w3ClosingPage=!url`

This message indicates an application setup error and not a run-time error, and thus should not be visible to the user during normal operations. It is not kept in the system messages file.

ERROR - Invalid or stale cookie(000)

This message displays to the user when a previously established connection is no longer valid. This happens when a connection times out or when the user attempts to reload a cached page. Restart the connection to recover from this state.

"xxxx" is an illegal command

This message displays if the application name specified in the w3exec parameter does not exist in the w3Apps file. Assuming the application name is app, this URL example shows the error:

`http://w3lib.rainingdata.com/cgi-bin/fccgi.exe?w3exec=wrongapp&w3ClosingPage=!url`

This URL example shows the correction:

`http://w3lib.rainingdata.com/cgi-bin/fccgi.exe?w3exec=app&w3ClosingPage=!url`

This message indicates an application setup error and not a run-time error, and thus should not be visible to the user during normal operations. It is not kept in the system messages file.

Socket message

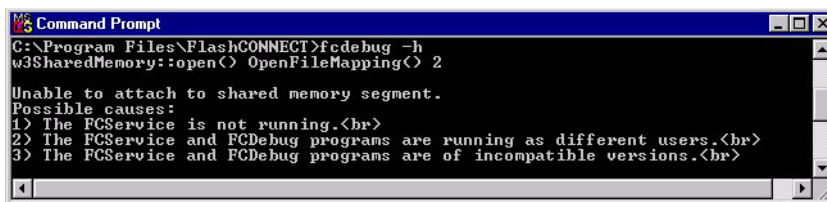
- Verify that the Web server and the database component are both running compatible versions of FlashCONNECT.
- **For D3:** If `WWW-STATUS` shows all lines with a status of sockets, verify that D3 7.1 or later is running.

Windows Platform Error Messages

Error 1067

If starting the FlashCONNECT server on the Web server host produces this error, shut down and restart the server.

Unable to attach to shared memory segment.



Please refer to the diagnostic message [Unable to connect to FCService in Diagnostic Messages Displayed On the Browser.](#)

Failed to open semaphore

There is a version mismatch between fccgi.exe and FCService. The versions must match. Reinstalling both the D3 and Windows components may be necessary. To determine the version of fccgi.exe and FCService refer to the topic [Determining Release Levels](#). (Entering -v from MS-DOS no longer performs this function.)

Sorry no FlashCONNECT servers are available at this time

Try again in a few moments. There are no available FlashCONNECT connections. Possible causes:

- There are no connections started.
- No phantoms are available.
- No database licenses are available.
- They are all in use.

ERROR - Missing required elements in form

This message displays if the URL starting a FlashCONNECT application is missing either a w3Exec parameter or a w3SessionId parameter. This also happens if the calls to w3HtmlFormBeg or w3HtmlFormEnd are missed.

This URL shows the error:

<http://w3lib.rainingdata.com/cgi-bin/fccgi.exe?w3ClosingPage=!url>

This URL shows the correction:

`http://w3lib.rainingdata.com/cgi-bin/fccgi.exe?w3exec=app&w3ClosingPage=!url`

This message indicates an application setup error and not a run-time error, and thus should not be visible to the user during normal operations. It is not kept in the system messages file.

ERROR - Invalid or stale cookie(000)

This message displays to the user when a previously established connection is no longer valid. This happens when a connection times out or when the user attempts to reload a cached page. Restart the connection to recover from this state.

"xxxx" is an illegal command

This message displays if the application name specified in the w3exec parameter does not exist in the w3Apps file. Assuming the application name is app, this example shows the error:

`http://w3lib.rainingdata.com/cgi-bin/fccgi.exe?w3exec=wrongapp&w3ClosingPage=!url`

This example shows the correction:

`http://w3lib.rainingdata.com/cgi-bin/fccgi.exe?w3exec=app&w3ClosingPage=!url`

This message indicates an application setup error and not a run-time error, and thus should not be visible to the user during normal operations. It is not kept in the system messages file.

Socket message

- Verify that the Web server and the database component are both running compatible versions of FlashCONNECT.
- For D3: If `WWW-STATUS` shows all lines with a status of sockets, verify that D3 7.1 or later is running.

No FlashCONNECT Services available at this time

- If you do not specify w3ServerPool, verify that there is a default server pool.
- If you do specify a server pool, verify that there are ports available.
- Verify that all applications are *not* going to the debugger.
- Verify that all applications return to `WWW-IRUN` and do *not* end.
- Check to see if all ports are consumed by applications using persistent connections. If all ports are consumed, either modify the application to *not* use persistent connections, or allocate more ports.

Customer Service

Raining Data Customer Service representatives are available to provide assistance. For current information, please refer to the Raining Data Web site at:

<http://www.rainingdata.com/support/cservice/customer.html>

Contacting Customer Service

Refer to the topics below for more information.

Providing Required System Information

When you contact any of our offices for support, please have the System ID available. This number is located on the back of the product CD package.

Receiving Extended Support Services

Extended support services are available, preferably with advance notice. Extended hours are those hours outside our standard business hours, which includes holiday closures. Please call the Raining Data office in the United States at 1-949-261-1875, Monday through Friday, to make special arrangements.

Holiday Closures

Holiday closures are subject to change. Raining Data administrative offices in the United States are closed during most national holidays. Please see the Raining Data Web site for complete holiday information of your nearest office at:

<http://www.rainingdata.com/support/cservice/locations.html>

Purchasing Support Packages

Support packages, including software upgrades, are available. Contact a Raining Data sales representative at 1-949-442-4400 for details.

If there are problems installing this product, please contact your customer service representative.

Activating Your Software

Activations are available 24 hours a day, seven days a week at the Raining Data office in the United States at 1-949-261-1875 or through the World Wide Web at:

www.rainingdata.com/support/activation/activation.html

Index

Symbols

- /etc/inittab 86, 92
- ? FCdebug command 147
- [] brackets v
- { } braces v
- | pipe sign v
- | vertical line v
- |-V FCDebug command 150

A

- ABS base
 - WWW-STATUS 98
- adding
 - FlashCONNECT Socket servers 62
- ADI 43
- administering
 - log files 110
- administration reports 49
 - completed sessions 51
 - concurrent users 50
 - contacts 52
 - line status 55
 - redirect 53
 - rotating content use 54
 - usage 52
- administration utility
 - environment tab 89
 - startup tab 90
 - Windows 89
- administrative tasks 26
- Anonymous
 - applications 37
- Apache Web server 12, 104
 - customizing error messages 104
- API (Application Program Interface) 20
- applications 29
 - adding new applications 35
 - adding users and groups 35
 - debug strategies 146
 - deleting 37
 - dynamic 10, 13
 - editing 35
 - interactive 16

- static 10, 13
- timeout field 37
- using secured 25
- viewing FCServlet diagnostic 84
- WWW-STATUS 98
- architecture
 - FlashCONNECT 11
- authentication process 27
- automatic load balancing 46
 - example 47
 - implementing 46

B

- binary data 115
- bold 5
- boo application
 - Loop-Back Test 143
- braces { } v
- brackets [] v
- browser
 - diagnostic messages 155
 - server error 206
 - verifying access 171
 - Web 7
- browser link 17

C

- category field
 - mvDesigner 37
- CGI 8
 - interface 12
 - Web server 104
- CGI/Environment variable 155
- cgi-bin directory 22
- Change Account Failed error
 - socket server 67
- character translation 115
 - deleting server pools 119
 - error messages 120
 - one to one 116
 - one to two 116
 - two to one 116
 - w3ProfileId 117

- w3ServerPool 117
- w3SessionId 117
- Character Translation component
 - enabling 117
- Checkup Program
 - D3 version 140
 - diagnosing FlashCONNECT problems 138
 - FC Host port 140
 - FlashCONNECT activation 140
 - licenses 141
 - licenses available 140
 - phantom processes available 140
 - phantoms 141
 - valid host names 142
- clearing phantom entries 112
- Closing Page Sent error
 - socket server 67
- completed sessions report 51
- components
 - Character Translation 117
 - configuring 85
 - Java 71
 - starting 85
 - stopping 85
- concurrent users report 50
- configuration
 - Checkup Program diagnostic 138
 - D3 system 86
 - diagnosing problems 138
 - host 86
 - limits 48
 - server pool connections 48
 - startup 44
 - turnkey 86
 - unknown line in configuration file 206
 - WWW-START 44
- configuring
 - components 85
 - heartbeat 102, 103
- connecting
 - server pools 42
- connection pooling 39
- connections
 - configure server pools 48
 - FlashCONNECT server and Web server using WWW-STATUS 96
 - server pooling 39
 - terminating FlashCONNECT socket server 64
- contact customer service 212
- contacts reports 52
- conventions
 - document 4
 - syntax iv
- Cookie 149
- cookie
 - invalid 206
 - stale 206
- cookies 14
- customizing
 - FlashCONNECT server 95
 - Web server 95
 - WWW-IRUN 103
- D**
- D3
 - TCL 26
- D3ServerSlots 147
- database
 - server 20
- date, entering 5
- debug
 - command 146
 - recording incoming variables 151
 - recording outgoing HTML code 151
 - server debug program 147
 - server pools 150
 - step-by-step 146
 - w3Logs 146
 - w3Monitor 151
- Debug Mode
 - applications 37
- debug strategies
 - applications 146
 - dynamic applications 146
 - static applications 146
- debugger 147
 - tandem 144
- deleting phantom entries 112
- determine release level 157
- diagnosing
 - problems 137, 159, 169, 181
- diagnostic 71
 - fccgi.exe 174, 175
 - fccgi.exe can access D3 server 177
 - FCService 175
 - FCService and D3 server are connected 176
 - FCServlet 78, 83
 - Loop-Back Testw3logs, loopback file 143

- PERL scripts 173
- verify browser access 171
- verify browser access to static Web pages 171
- Web server scripts running in cgi-bin directory 172
- diagnostics
 - messages displayed on browser 155
 - UNIX 160
 - Windows platform 170
- directories
 - cgi-bin 22
 - Inetpub 22
 - scripts 22
 - virtual 22
 - w3library 22
 - WebContent 22
- directory
 - virtual 10
- dm users file 32
- document conventions 4
- drivers
 - program 103
 - WWW-IRUN 103
- drivers, WWW-IRUN 20
- dynamic application 10, 13, 19, 27
 - adding users and groups 35
 - authentication process 27
 - debug strategies 146
- dynamic page
 - authentication process 28
- E**
- e-commerce 9
- entering data 5
- env_vars_sema 147
- environment
 - customizing local 89
 - tab 89
 - variable name space 91
 - variables 91
- ENVSize 147
- error codes
 - FlashCONNECT Socket Server 66
- error messages
 - error 1067 208
 - failed to open semaphore 208
 - FlashCONNECT Event Log entries 178
 - FlashCONNECT Socket Server 66
 - for Web servers 104
 - illegal command 206
 - invalid or stale cookie 206
 - missing required elements in form 206
 - shared memory segment 208
 - socket message (UNIX) 207
 - socket message (Windows) 209
 - unknown line in configuration file 206
 - Windows platform 208
- errors
 - character translation messages 120
- escape character 118
- Event Log
 - entries 178
 - heartbeats 101
- F**
- fatal errors
 - socket server 67
- fccgi.exe
 - diagnostic 174, 175
 - set location 104
 - UNIX 19
 - Web server 21
 - Windows 19
- FCConfig.exe 117
- FCdebug 147
- FCdebug.000n file 153
- FCHeartbeat 147
- fcisapi.dll 19
 - set location 104
- FCIsUp 148
- FCPort 147
- FCService
 - connection 19
 - diagnostic 175
 - diagnostic messages 155
 - diagram 18, 19
 - heartbeat 102
 - installation 92
 - sending a heartbeat 102
 - starting 92
 - stopping 93
 - w3exec 21
- FCServicePID 148
- FCServlet 21, 71
 - diagnostic 78
 - diagnostic reports 83
 - explanation 73
 - overview 72

- reconfiguring diagnostic parameters 83
- running diagnostics from FlashCONNECT
 - Socket Server 78
- setting diagnostic parameters 78
- testing 74, 81
- viewing application information 84
- FCServlet diagnostic 71
- fcss.jar 58
- FCXlate 117
- FCXlate parameter 117
- files
 - JOBS 112
- firewall
 - heartbeats 101
- FlashCONNECT
 - overview 7
- FlashCONNECT components 85
 - diagram 11
- FlashCONNECT Logs Control Maintenance 110
- FlashCONNECT Maintenance Pages 30
- FlashCONNECT Maintenance Web pages 26
- FlashCONNECT server
 - customizing 95
- FlashCONNECT Socket Server
 - accessing 58
 - changing location of log file 66
 - clearing monitor information 70
 - error codes 66
 - FCServlet diagnostic tests 78
 - log 65
 - monitor information 67
 - terminating connections 64
 - understanding log information 69
- FlashCONNECT Socket Server Manager 26, 57
 - exiting 70
 - starting 59
- flashconnect.conf 117
- flashconnect.log 120
- form variables 155
- frequency
 - heartbeat 102

G

- groups 29
 - adding 32
 - adding from FlashCONNECT Maintenance Web page 32
 - adding users to 33
 - assigning users to dynamic applications 35

- deleting 33
- deleting users from 34
- maintaining 29

- GUI 9

H

- h FCdebug command 147
- heartbeat
 - configuring 102, 103
 - FCService 102
- heartbeat frequency 102
- Heartbeat Period 91
- heartbeat period 102
- heartbeats
 - Event Log 101
 - firewall 101
- Host
 - WWW-STATUS 97
- Host Name 149
- HTML (hypertext markup language) 16
- HTTP (hypertext transfer protocol) 16
- HTTPD server 14, 19

I

- I/O error
 - socket server 67
- Ident
 - WWW-STATUS 97
- Idx
 - WWW-STATUS 97
- IIS 104
- IIS server
 - customizing error messages 104
- IIS Web server 12
- Incorrect w3Exec, w3SessionId error
 - socket server 67
- individual lines
 - stopping 41
 - stopping server pools 41
- Inetpub directory 22
- installation
 - turnkey 86
- interactive application 16, 17
- Internet
- Internet Information Server 104
- Invalid Session error
 - socket server 66
- IP address

- socket server 64
- ISAPI 8
- ISAPI interface
 - web server 104
- italic font style iv

J

- Java components 10, 71
 - testing with FCServlet diagnostic 82
- Java Native Interface 71
 - overview 72
- Java Runtime Environment 10
- Java Servlet 8
- Java Servlet Engine 10
- Java Servlet engine 72
- Java servlets 10
- Java Virtual Machine 58
- Javascript 9
- JNI
 - Java Native Interface 71
- JNI interface 72
- JOBS file 112

K

- kill client 64
- kpool
 - FCDebug command 148

L

- letters representing numbers v
- Line
 - WWW-STATUS 98
- line status report 55
- load balancing 10
- local environment
 - customizing 89
- log
 - FlashCONNECT Socket Server 65
- log file
 - changing FlashCONNECT Socket Server 66
 - w3Logs,Dump 111
 - w3Logs,errors 111
 - w3Logs,FlashCONNECT 111
 - w3Logs,w3Rotator 111
- log files
 - administering 110
- logging on

- FlashCONNECT Maintenance Pages 30
 - timeout 31
- Loop-Back Test
 - application diagnostic 143

M

- Macromedia Dreamweaver 9
- maintaining
 - applications 29
- maintenance pages
 - logging on 30
 - password 30
 - user name 30
- Maintenance Web pages 26
- manually controlling server pools 48
- Maximum Message Length Exceeded error
 - socket server 67
- maximum servers supported 91
- mcompile command 103
- membership
 - adding and removing users 34
- messages
 - character translation 120
 - customizing error messages for Web servers 104
- Microsoft Front Page 9
- middleware 8
- monitor information
 - clearing 70
 - viewing FlashCONNECT Socket Server 67
- monospace
 - bold font style iv
 - font style iv
- monospace bold 5
- multiuser interactive application 17
- mvDesginer
 - socket server connection 57
- mvDesigner
 - graphical user interface 9
- mvDesigner server 8, 10

N

- named pipes 8
- Netscape server
 - customizing error messages 104
- Netscape Web server 12, 104
- network connection
 - heartbeats 101

No Connections error

 socket server 66

No Response error

 socket server 66

No Such Application error

 socket server 66

non-ANSI data 115

NT See Windows

P

parameter

 FCXlate 117

password

 maintenance pages 30

PCB FID

 WWW-STATUS 98

PERL 173

PERL scripts

 executing 163

persistent connections

 item locks 13

 peripheral devices 13

 record locks 13

persistent data

 BASIC application 13

 Common application 13

 dynamic application 13

 resources 13

phantom entries 112

phantom process

 tandem 146

PIB Stat

 WWW-STATUS 98

Pick BASIC business rules 8

Pipe Name 149

pipe sign, | v

pNN

 FCDebug command 150

pooling

 connection 39

 server 39

Port

 WWW-STATUS 97

problems

 diagnosing FlashCONNECT configuration 138

Profile ID

 applications 36

R

R1 & Return stack contents

 WWW-STATUS 98

redirect report 53

release level, determine 157

reports

 completed sessions 51

 concurrent users 50

 contacts 52

 FCServlet 83

 line status 55

 redirect 53

 rotating content use 54

 usage 52

 Web-based 49

resource allocation 10

resources

 BASIC application 13

 item locks 13

 memory 13

 peripheral devices 13

 record locks 13

revision history 157

rotating content use report 54

S

scripts 22

scripts directory 22

Secure Socket Layer (SSL)

 applications 36

secure socket server layers 10

secured applications 25

security

 secure socket server layers 10

select Web server 12

Sema

 semaphore number 148

semaphores 208

Semaphores per ID 148

server

 debug program 147

 HTTPD 19

 maximum supported 91

 mvDesigner 10

 table entries 148

 Web 10

server administration

 Windows platform 104

-
- server error 206
 - server pool
 - name 43
 - server pooling 39
 - introduction 39
 - server pools 39
 - administering 39
 - assigning connection 45
 - automatic load balancing 46
 - configuring connections 48
 - connecting 42
 - debugging 150
 - deleting translation server pools 119
 - diagnostics 166, 176
 - groups 40
 - individual lines 40
 - manually controlling 48
 - specifying the relationship 44
 - starting 40
 - stopping 41
 - service 92
 - environment tab 89
 - port 91
 - startup tab 90
 - TCP/IP port 91
 - services
 - socket server 60
 - sessions
 - beginning and ending 16
 - tracking 17
 - sessions report 51
 - shared memory segment error 208
 - Since
 - WWW-STATUS 98
 - socket message 207, 209
 - socket server
 - accessing 58
 - clearing monitor information 70
 - exiting 70
 - fatal errors 67
 - fcss.jar file 58
 - FlashCONNECT manager 57
 - kill client connection 64
 - mvDesigner 57
 - starting 59
 - terminate connections 64
 - terminating connections 64
 - understanding log information 69
 - Windows platform services 60
 - Socket Server Manager 26
 - socket servers
 - adding FlashCONNECT socket servers 62
 - stopping FlashCONNECT socket servers 63
 - Sockets
 - WWW-STATUS 97
 - starting
 - components 85
 - FlashCONNECT Socket Server Manager 59
 - server pools 40
 - starting FlashCONNECT on D3 93
 - startup tab 90
 - environment variable name space 91
 - FlashCONNECT service port 91
 - maximum servers supported 91
 - pause between searches 91
 - searches before server unavailable status 91
 - Stat
 - WWW-STATUS 98
 - State
 - transitional 148
 - static
 - application 13
 - page definition 13
 - static application 10
 - authentication process 27
 - debug strategies 146
 - static page
 - authentication process 27
 - static Web pages
 - diagnostic 171
 - Status
 - WWW-STATUS 98
 - status messages
 - w3Lines 98
 - WWW-STATUS 98
 - stopping
 - components 85
 - FlashCONNECT socket servers 63
 - server pools 41
 - stopping FlashCONNECT
 - Windows, WWW-STOP 41
 - stopping FlashCONNECT on D3 94
 - stopping server pools 41
 - syntax conventions iv
- T**
- t
- FCDebug command 148

- tandem 146
 - debugger 144
 - example code 144
- TCP/IP connection 8
- TCP/IP port 91
- TCP/IP Port number
 - socket server 63
- TCP/IP port number
 - socket server 58
- terminating
 - socket server connections 64
- testing
 - FCServlet 74, 81
- TimeOut 149
- timeout
 - applications 37
 - FlashCONNECT Web pages 31
- tr
 - FCDebug command 149
- translation
 - character translation 115
 - deleting server pools 119
 - errors 120
 - one to one 116
- translation files 117
- translation hex pair 118
- translation map 118
- troubleshooting 137, 159, 169, 181
 - debug strategies 146
 - UNIX platform 160
- troubleshooting diagnostics
 - UNIX 160
 - Windows platform 170
- Troubleshooting the Windows Platform 170
- turnkey installation 86

U

- Unauthorized Access error
 - socket server 66
- Undecipherable Message error
 - socket server 66
- understanding log information
 - socket server 69
- unicode data 117
- UNIX
 - administrative tasks 26
 - fccgi.exe 19
 - systems supported iv
 - turnkey installation 86, 92

- Web server 12
- UNIX platform
 - diagnostics 160
- UNIX platforms
 - troubleshooting 160
- Unknown Protocol error
 - socket server 67
- uNN
 - FCDebug command 150
- unnamed common 103
- Unsupported Feature error
 - socket server 67
- URL
 - error message 206
- usage reports 52
- User
 - WWW-STATUS 97
- user name
 - maintenance pages 30
- user-coldstart macro 44, 86
- users 29
 - adding and removing 34
 - adding to groups 33
 - assigning to dynamic applications 35
 - deleting from groups 34
 - maintaining 29
- users and groups
 - w3apps file 29
- user-shutdown 86

V

- v
 - FCDebug command 150
- version 157
- vertical line, | v
- virtual directories 10
- virtual directory 22

W

- w3Apps 27, 32
- w3Apps file 35
 - deleting applications 37
- w3apps file
 - users and groups 29
- w3Config 103
- w3Debugging tool 152
- w3exec 21
- w3Groups 27, 32

- w3HostName
 - see w3ServerPool 43
- w3Html
 - static vs. active pages 14
- w3HtmlFormBeg 206
- w3HtmlFormEnd 206
- w3library 22
- w3library directory 22
- w3lines 43
- w3Lines status 98
- w3Log subroutine 110, 146
- w3Logs 110, 146
 - debug 146
- w3Logs,Dump
 - log file 111
- w3Logs,Dump file 151
- w3Logs,errors
 - log file 111
- w3Logs,FlashCONNECT
 - log file 111
- w3Logs,w3Rotator
 - log file 111
- w3Logs.Control 110
- w3Logs.control file 111
- w3Monitor 40
 - automatic load balancing 46
 - debugging 151
 - manually controlling 48
 - starting 94
 - stopping 94
- w3Monitor frequency 102
- w3ServerPool 43, 45
- w3Tandem 144
- w3tandem example 144
- w3Users 27, 32
- w3Users file
 - adding users to groups 33
- Web
 - applications 16
 - tracking state 13
- Web browser 7
 - verifying access 171
- Web content directory 22
- Web page 8
- Web server 10
 - Apache 12
 - customizing 95
 - fcgi.exe 21
 - fcisapi.dll 21
 - FCServlet 21
 - Netscape 12
 - selecting 12
 - Windows platform 104
- Web servers
 - Apache 104
 - customizing Web server error messages 104
 - IIS 104
 - Netscape 104
- WebContent directory 22
- where command 98
- Windows
 - 2000 iv
 - administration utility 89
 - administrative tasks 26
 - fcgi.exe 19
 - fcisapi.dll 19
 - FlashCONNECT administration 104
 - requirement iv
 - server debug program 147
 - Web server 12
 - XP Professional iv
- Windows platform
 - diagnostics 170
 - environment variables 89
 - error messages 208
 - FlashCONNECT service 92
 - startup tab 90
 - troubleshooting 170
 - troubleshooting diagnostics 170
 - turnkey installation 86
 - Web server 104
- Worker ID
 - socket server 64
- WorkerID
 - socket server 64
- World Wide Web 16
- WWW-CONFIGURE 44, 103
- WWW-CONTINUE 103
- WWW-IRUN 40, 103
 - customizing 103
 - FlashCONNECT driver 20
 - unnamed common 103
 - w3exec 21
- WWW.IRUN2 103
- WWW-MANAGER 40, 48
- WWW-MANAGER START 86
- WWW-MANAGER STOP 86
- WWW-MANAGER w3Monitor 41

WWW-POOL 40
WWW-START
 configuration 44
WWW-STATUS 96, 98, 207
 line descriptions 97
 server 96
WWW-STOP 41
WWW-TANDEM 144

X

X windows 61